DETERMININING THE AIRCRAFT REQUIREMENT FOR TURKISH AIR MOBILITY SYSTEM

THESIS

Mehmet M Arı, First Lieutenant, TUAF

AFIT/GOR/ENS/00M-01

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2000	Master's Thesis

4. TITLE AND SUBTITLE

DETERMINING THE AIRCRAFT REQUIREMENT FOR
TURKISH AIR MOBILITY SYSTEM

5. FUNDING NUMBERS

6. AUTHOR(S)

Mehmet M. Ari 1st. Lt. TUAF

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOR/ENS/00M-01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Hava Kuvvetleri Komutanlığı
Egitim Daire Baskanligi
Bakanlıklar 06100 Ankara/ Turkey
Tel: +90-312-4172150

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Dr. James T. Moore, Room 185, Phone (937) 255-6565 ext. 4337, E-mail: james.moore@afit.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

ABSTRACT (*Maximum 200 Words*)

Aircraft technology develops very rapidly. To keep up with the modern technology, we should update our systems and material over time. In order to do so, we should keep track of modern technologies, and implement them in our military. Turkish Air Force Command operates a vital logistical system for which routing and scheduling are done on a weekly basis. This research reviewed weekly schedules and determined the average required weekly missions, which would be flown during the week. By carefully examining the job – machine formulation, this research devised a method for determining the aircraft required to perform for weekly missions. This method could help determine aircraft requirements for procurement. If future research can improve this method using the recommendations provided, this method could become a part of TAMC's significant planning process.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Scheduling Problem, Air Mobility Problem, Integer Programming, , Turkish Air Force, Turkey	111
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

AFIT/ENS/GOR/00M-01

AFIT/ENS/GOR/00M-01

DETERMININING THE AIRCRAFT REQUIREMENT FOR TURKISH AIR MOBILITY SYSTEM

THESIS

MEHMET M. ARI, 1ST LT.

AFIT/ENS/GOR/00M-01

AFIT/ENS/GOR/00M-01

DETERMINING THE AIRCRAFT REQUIREMENT FOR TURKISH AIR MOBILITY SYSTEM

THESIS

Presented to the Faculty of the Graduate School of

Engineering and Management

Air Education and Training Command

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Mehmet M. Arı, B.S.

1st Lieutenant, TUAF

March 2000

AFIT/ENS/GOR/00M-01

DETERMINING THE AIRCRAFT REQUIREMENT FOR TURKISH AIR MOBILITY SYSTEM

Mehmet M. Arı, B.S.
First Lieutenant, TUAF

Approved:

James T. Moore — 9 Mar 00

James T. Moore (Advisor) — date

Mark A. Gallagher — 10 Mar 00

Mark A. Gallagher (Reader) — date

Acknowledgement

I would like to express my appreciation to my thesis advisor, Dr. James T. Moore for taking the challenge of guiding me through this thesis effort. His guidance and support throughout this effort were invaluable. His experience and knowledge helped me to progress and finish my thesis.

I would also like to thank Mr. and Mrs. Grazier for their support, they made me part of their family, and I am forever grateful.

I owe also a great deal of thanks to the Turkish Air Force for providing me this wonderful Master's program opportunity. I will deserve it by working harder for my country.

Most importantly, I would like to thank my family for raising and supporting me all my life. My gratitude will be forever.

Mehmet M Arı

Table of Contents

Figures

Tables

Abstract

Aircraft technology develops very rapidly. To keep up with the modern technology, we should update our systems and material over time. In order to do so, we should keep track of modern technologies, and implement them in our military.

Turkish Air Force Command operates a vital logistical system for which routing and scheduling are done on a weekly basis. This research reviewed weekly schedules and determined the average required weekly missions, which would be flown during the week.

By carefully examining the job – machine formulation, this research devised a method for determining the aircraft required to perform for weekly missions. This method could help determine aircraft requirements for procurement. If future research can improve this method using the recommendations provided, this method could become a part of TAMC's significant planning process.

I. CHAPTER I

I.1 RELEVANT INFORMATION ABOUT TURKEY

In this chapter, I describe my research. Section I.1 gives general information on the geographical and political history of Turkey. Section I.2 gives insight about my research problem.

I.1.1 THE TURKISH REFERENCE LIBRARY

The location of Turkey is at a place where the three continents making up the old world, Asia, Africa and Europe, are closest to each other. Turkey actually is at the point where Europe and Asia meet.

Geographically, Turkey is a large, roughly rectangular peninsula situated bridge-like between southeastern Europe and Asia. Indeed, the country has functioned as a bridge for human movement throughout history. Turkey extends more than 1,600 kilometers from west to east but generally less than 600 kilometers from north to south. Total land area is about 814,578 square kilometers, of which 790,200 square kilometers are in Asia and 24,378 square kilometers in Europe. The country is located in the northern half of the hemisphere at a place that is about halfway between the equator and the north pole, a latitude of 36 degrees N to 42 degrees N and a longitude of 26 degrees E to 45 degrees E.

Two European and six Asian countries are surrounding the land of Turkey as neighbor. The land border to the northeast with the Commonwealth of Independent States is 610 kilometers long, that with Iran, is 454 kilometers long, and that with Iraq is 331 kilometers long. In the south is the 877 kilometer-long border with Syria. Turkey's borders on the European continent consist of a 212-kilometer frontier with Greece and a 269-kilometer border with Bulgaria.

Figure I.1. Geographic Map of Turkey (Discover Turkey, 2000)

Because of its geographical location, the mainland of Anatolia (Turkey) is a bridge connecting the Middle East and Europe, and it shares in the history of both those parts of the world. Despite the diversity of its peoples and their cultures, and the constantly shifting borders of its ethnic map, Anatolia has a history characterized by

remarkable continuity. It is the birthplace of many great civilizations. It has also been known as a center of commerce because of its land connections to three continents and the seas surrounding it on three sides. (Embassy of the Republic of Turkey, 2000)

I.1.2 TURKISH POLITICS & POLICY

The government of Turkey functions in accordance with the constitution of 1982. The Republic of Turkey is a democratic, secular and social state governed by the rule of law respecting human rights and loyal to the political philosophy of Kemal Atatürk., who was the Republic's founding father. The Turkish State, with its territory and nation, is an indivisible entity.

Turkey is fully committed to democracy, respect for human rights and fundamental freedoms, the rule of law and a free market economy. Closer integration into the international community, in particular into Western institutions, has always been a priority of Turkish foreign policy.

Education is based on contemporary science and education methods and is provided under the supervision and control of the state. The official language of the Turkish State is Turkish and its capital is Ankara. Everyone possesses inherent fundamental rights and freedoms. The individual is entitled to privacy and to freedom of thought and communication

The fundamental goals and duties of the State are to safeguard the independence and the integrity of the Turkish Nation. (Embassy of the Republic of Turkey, 2000 Embassy of the Republic of Turkey, 2000)

I.1.3 RELATIONS WITH THE UNITED STATES OF AMERICA

During the postwar era, Turkey's foremost ally has been the United States. Because of Turkey's strategic location in the Middle East, its proximity to the Soviet Union's military installations and test sites, and its control of the Black Sea straits, military ties with the United States were a crucial factor in the East-West confrontation. The alliance originated soon after the end of World War II. The Truman Doctrine of 1947 was the beginning of a new era in Turkish-American relations. Close working relations were established between Turkey and the US in the political, military, economic, technical, social and cultural affairs during this period.

In accordance with bilateral defense arrangements under NATO authority, the United States has developed and maintained several major military installations on Turkish bases. (Republic of Turkey Ministry of Foreign Affairs, 2000)

> Bilateral relations faced certain hardships during the early 1960's and mid 1970's. However, whatever problems were encountered during this period, they never jeopardized the strongly underlying partnership between the two countries. A new chapter in Turkish-American relations opened in the 1980s. Greater cooperation and US support for Turkey increased significantly.
> In 1991 Turkey and the US agreed to upgrade their cooperation even further and give it the status of an Enhanced Partnership. Since then bilateral relations have continued to prosper in many fields. Post-Cold War developments have clearly shown that more than ever Turkey and the US currently share a set of common strategic, security and economic concerns and interests which naturally bring them closer together. (Republic of Turkey Ministry of Foreign Affairs, 2000)

I.2 INTRODUCTION

I.2.1 BACKGROUND

Because of Turkey's location and its political and military ties, it maintains a strong vibrant military force. A vital function of its military is the movement of people and cargo. Turkish Air Mobility Command operates a vital logistical system for which routing and scheduling are done on a weekly and monthly basis. This system is referred to as a channel system. A channel consists of origin and destination airbases, known as an Origin – Destination (O – D) pair, between which Air Mobility Command provides regularly scheduled airlift. Channels are established in response to various demands, such as the pickup and delivery of cargo or passengers.

Flexibility within the system allows for either direct connections between Origin – Destination pairs, where a cargo airlifter would fly direct from the origin airbase to the destination airbase, or may entail service with several intermediate stops before arriving at the destination airbase.

In addition, certain O – D pairs may not be serviced by a single route which connects the two bases. In this circumstance, transshipping is required.

There are two other important definitions:

1. Frequency channels, such as those missions typically flown to special places, are scheduled at specified intervals and are not driven by cargo requirements.
2. Requirement channels are routes flown between O – D pairs with the number of missions flown based on the amount of cargo required to be transported.

In analyzing such a problem, the following decision variables need to be specified:

- Number of transport aircraft that are needed.
- Transport aircraft routes.

In the USAF Air Mobility System, a backup has been established in the event that Air Mobility Command's (AMC's) assets cannot deliver all required cargo. This backup is the Civil Reserve Air Fleet (CRAF), consisting of civilian commercial transportation that is contracted as necessary by AMC's Tanker Airlift Control Center (TACC). The CRAF is an essential, significant source of augmentation for the channel system on an ongoing basis. However, the Turkish Air Mobility System does not include a Civil Reserve Air Fleet component.

I.2.2 PROBLEM STATEMENT

The problem addressed by this study is presented in the following question. Given, there is only one type of cargo aircraft available, a set of routes, and demand between O – D pairs, what is the minimum number of aircraft required to satisfy this requirement? Answering this question would help the Turkish Air Mobility Command make good decisions when purchasing a new type of cargo aircraft. The results would show them how many aircraft they need to accomplish their mission.

Besides the O-D pairs, there are also some weekly requirements to serve passengers. Planes provide transportation for passengers and extra cargo. However, the flights are not scheduled according to passenger demand. The routes the planes fly can

be examined in terms of total distance and there may be some improvements made in this set of routes.

I.2.3 ASSUMPTIONS AND SCOPE

The following simplifying assumptions are made:

- This research assumes that the cargo requirements for all Origin – Destination pairs are known deterministically. (i. e. with certainty). Turkish Air Mobility Command analysts forecast these cargo requirements based on weekly/monthly trends.
- Cargo is classified by weight only; therefore the cargo can be divided into an infinite number of subsets. Any other characteristics, such as size and priority, are assumed to be the same for all cargo. Any mixture of cargo is allowed on a single aircraft.
- The capacity of the aircraft is known.
- There are no load balance or size restrictions. That is, any part of the cargo can fit into any part of the aircraft as long as the maximum takeoff weight restriction is not violated. This is reasonable in most cases because almost all pallets can fit anywhere in the transport aircraft except possibly in the rearmost part of the aircraft.
- Airbases are assumed to be capable of handling an unlimited amount of cargo (i.e., no restrictions on loading equipment or storage areas).
- Airbases are assumed to be available 24 hours a day.
- Passenger requirements are not considered and do not affect aircraft cargo capacity.

- The service time used at a base is:
  - 30 +/- 3 minutes: Non – maintenance ground operations to prepare an airlifter for flight (e.g. refueling and checklist operations)
  - 28 +/- 2.5 minutes: Loading/unloading time.
- Training missions can be postponed, if it is necessary.
- The maximum availability of an aircraft is considered as thirty hours a week.
- In the computations, only operational numbers of aircraft are taken into account. Maintenance loss is not considered.

I.2.4 RESEARCH OBJECTIVES

The purpose of this research is to determine the number of aircraft Turkish Air Mobility Command needs to successfully accomplish their mission. Additionally, another objective is finding the best tool for solving the problem as formulated.

CHAPTER II

II. LITERATURE REVIEW

SCOPE AND ORGANIZATION OF THE REVIEW

Given the background concerning the needs of Turkish Air Mobility Command in terms of the number of aircraft to cover certain missions, this review briefly discusses some previous thesis efforts in Section II.1. Section II.2 is about general concepts that give ideas about formulating a problem. Section II.3 presents information on scheduling, and Section II.4 tells what my formulation is based on.

II.1 PREVIOUS EFFORTS

Topcuoglu (1997) investigated ways to achieve the most efficient airlift system possible. He had operational experience in the Turkish Air Force as a cargo pilot, and he is familiar with the Turkish Air Mobility Command airlift problem. He reviewed the USAF Airlift System and investigated the applicability of various mobility models to the Turkish Air Mobility Command's airlift system. In conducting his research, he used Generalized Air Mobility Model (GAMM).

GAMM was developed by General Research Corporation (GRC) and the Directorate of Advanced System Analysis, Aeronautical System Center at Wright-

Patterson Air Force Base. GAMM is a Monte Carlo simulation of an airlift transportation system. The GAMM program is an event oriented simulation of the transportation system defined by the scenario and the cargo required to be moved. Topcuoglu presented a tactical mobiliy senario and develepod a thirty day schedule to cover the requirements of his scenario. He compared also the efficiency of the airplane types. Given cargo and frequency requirements, a fleet of aircraft and possible routes, he allocated aircraft to achieve the most efficient airlift system possible.

Moul (1992) investigated the cargo delay caused by a given mission schedule. In his research, he used a simplified route network to develop a method for measuring schedule effectiveness by determining the amounts of enroute cargo delay caused by a given mission schedule. He produced a computer simulation for measuring cargo delay.

The research by Del Rosario (1993) used a mathematical programming approach to flow cargo where a multiperiod, multicommodity network was used to model the channel cargo system.

The above research efforts are not similar to my problem since the number of aircraft was known. My first objective is to develop a model that can be used for finding the requirement of Turkish Air Mobility Command with respect to the number of aircraft.

II.2 GENERAL

Many researchers have studied the problem of finding optimal solutions to problems, which can be structured as a function of some *decision variables,* perhaps in

the presence of some *constraints.* The subject is very wide, and many books have already been written on its various aspects. Such problems can be formulated generally as follows:

$$\text{Minimize} \qquad f(x)$$

$$\begin{aligned} \text{subject to} \qquad & g_i(x) \geq b_i \quad ; \qquad i = 1, \ldots\ldots\ldots, m; \\ & h_j(x) = c_j \quad ; \qquad j = 1, \ldots\ldots\ldots, n; \end{aligned}$$

Here, x is a vector of decision variables, and general functions. (Actually the equality constraints are not strictly necessary, as they can be formulated in terms of pairs of inequalities, but it is often helpful to make them explicit.) This formulation has assumed the problem is one of minimization, but the modifications necessary for a maximization problem are clear.

There are many specific classes of such problems, obtained by placing restrictions on the type of functions under consideration, and on the values that the decision variables can take. Perhaps the most well-known of these classes is that obtained by restricting $f(.)$, $g_i(.)$ and $h_j(.)$ to be linear functions of decision variables which are allowed to take fractional **(continuous)** values, which leads to problems of linear programming. (Bazaraa et al, 1997: 7)

Combinatorial problems, have close links with linear programming (LP), and most of the early attempts to solve them used developments of LP methods. Often combinatorial problems are created by introducing integer variables taking the values 0

or 1, in order to produce a linear integer programming (IP) formulation. For example, in the case of the 0-1 knapsack problem, we define

$$\mathbf{x}_i \begin{cases} = 1 \text{ if item i is packed} \\ = 0 \text{ otherwise.} \end{cases}$$

The problem then reduces to the following integer program:

$$\text{Maximize} \quad \sum_{i=1}^{n} v_i x_i$$

$$\text{s.t} \quad \sum_{i=1}^{n} c_i x_i \leq C$$

c_i : units of capacity used by item i.

v_i : value of item i.

C : capacity.

However, although IP is often a difficult route finding optimal solutions to combinatorial problems, there are good reasons for its popularity. Mainly, the act of formulation is itself often helpful in defining more precisely the nature of a given problem. (Winston, 1994: 516)

II.3 SCHEDULING THEORY

II.3.1 WHAT IS SCHEDULING?

Scheduling is concerned with activities and resources. These certain activities are assigned to resources over certain time intervals. Assignment constraints must be satisfied and cost or "goodness" measures of the assignment should be optimized. The basic concepts of scheduling theory are defined below:

- *Activities:* A model of the activities can include their internal structure and characteristics, hierarchies of activity abstractions, and various operations on activities.
- *Resources:* A model of resources can include their internal structure and characteristics, hierarchies of resource abstractions, and various operations on resources.
- *Time:* A time model can include a calculus of time – points or time intervals.
- *Constraints:* A constraint model includes the language for stating constraints and a calculus for reasoning about them. Several classes of constraints commonly arise in practice. The most common are *precedence constraints* which state that one activity must precede another and *capacity constraints* which state bounds on the capacities of resources. A constraint calculus is used to analyze constraints and to propagate the effects of new constraints through a given constraint set.
- *Objectives:* Typically, the cost of a schedule is minimized. Cost can be measured in terms such as time to completion, work in progress, and total cost of consumed resources.
- *Scheduling problem:* Using the concepts above we can formulate a variety of scheduling problems. A reservation is a triple consisting of an activity, a resource, and a time interval. Generally, a schedule is a set of reservations that satisfy a collection of constraints and optimize (or produce a reasonably good value of) the objective. (Smith et al, 1995: 7-10)

II.3.2 THE GENERAL JOB SHOP SCHEDULING PROBLEM

Scheduling is "allocating resources over time to perform a collection of tasks" (Baker, 1974:2). The terminology of scheduling theory "arose in the processing and manufacturing industries" (French, 1982:5). The result is a standard description of a system in which the jobs are the collection of activities and the machines are the resources.

II.3.3 PARALLEL MACHINE MODELS

A series of parallel machines is important both theoretically and practically. From the theoretical viewpoint, these machines represent a generalization of the single machine, and appears much like flexible flow shop. From the practical standpoint, it is important because the occurrence of resources in parallel is common in the procedures for multistage systems.

Three principal objectives can be considered, such as the minimization of the makespan, the minimization of the total completion time, and the maximum lateness.

> With a single machine, the makespan objective is usually only of interest when there are sequence – dependent setup times; otherwise, the makespan is equal to the sum of the processing times and is independent of the sequence. When dealing with machines in parallel, the makespan becomes an objective of significant interest. In practice, one often has to deal with the problem of balancing the load on machines in parallel, and by minimizing the makespan the scheduler insures a good balance. (Pinedo, 1995: 61-62)

Scheduling parallel machines can be considered as a two – step process. First, determine which jobs should be allocated to which machines; later, determine the sequence of the jobs allocated to each machine. With the makespan objective, only the allocation process is important.

Preemption usually only plays a role when jobs are released at different times on a single machine. In contrast, on parallel machines, preemption plays more important role even when all jobs are released at the same time. As for my problem, preemption is not considered and is not allowable. The formulation of the problem will be discussed in Chapter 3. (Pinedo, 1995: 61-62)

The parallel machine job-scheduling model can be applied to my model with some changes and additional formulations. The linear programming formulation for P_m ll C_{max} can be the basis for my model. (P_m: m parallel machines; C_{max}: Makespan; the objective is to minimize maximum makespan). The next chapter explains the formulation and modeling of the research problem.

CHAPTER III

III. METHODOLOGY

Section III.1 describes other solution methods that can be used to solve this research problem. It also discusses advantages and weaknesses of these methods. Section III.2 is about the problem formulation. There is also a small example problem to make the concept clearer. In section III.3 and section III.4, the model restrictions and the solver are narrated.

III.1 SOLUTION METHODS

Like other problems, my problem can be solved with different methods. As for my problem, there are missions to be flown with the minimum number of aircraft. This means the research problem is similar to a job shop scheduling problem. We can assume aircraft are machines and missions are jobs. It can be illustrated as follows:

If we have m identical aircraft and n missions with different process times, we will assign the n missions to the m aircraft. Each mission is a single operation that takes space on the aircraft and can typically be processed on any aircraft. Suppose $m = 5$ and $n = 9$ and each mission has different processing time on each aircraft. We initially assign each mission to an aircraft. We define our move operation as the longest mission which can be moved to another aircraft.

Each aircraft has capacity up to 10 units.

J1 = 6 units

J2 = 4 units

J9 = 5 units

J3 = 8 units

J4 = 2 units

J5 = 5 units

J6 = 6 units

J7 = 3 units

J8 = 1 units

First missions are assigned arbitrarily to the aircraft. Then at each iteration a move is selected.

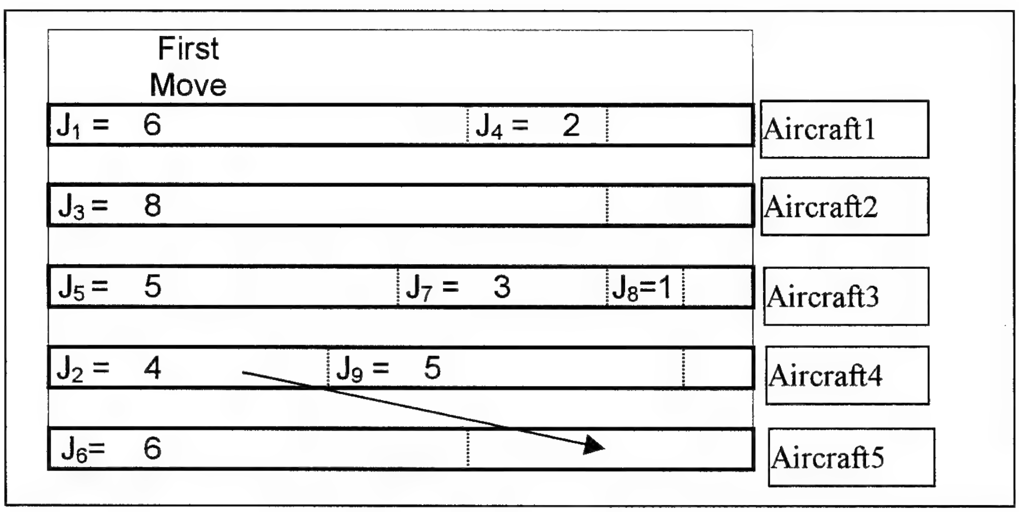


Figure III.1. First move

Mission 2 is moved from aircraft 4 to aircraft 5. Aircraft 5 has reached its capacity.

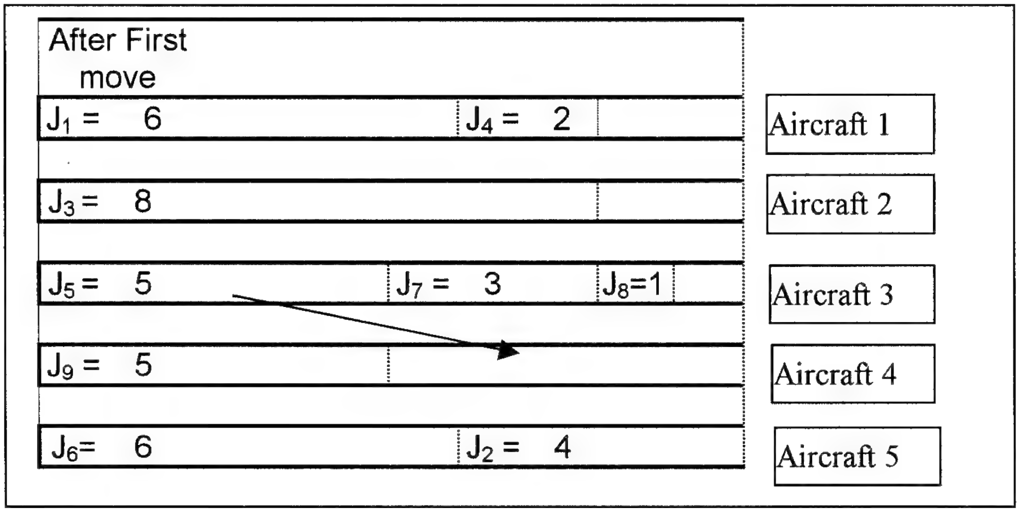


Figure III.2. After first move

Mission 5 is selected to be moved from aircraft 3 to aircraft 4. Aircraft 4 has reached its capacity.

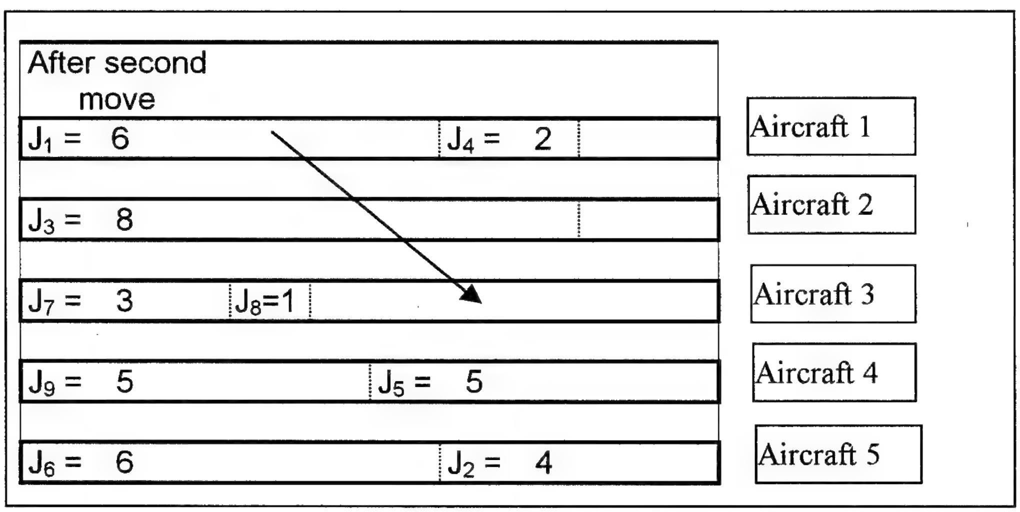


Figure III.3. After second move

Mission 1 is moved from aircraft 1 to aircraft 3. Aircraft 3 has reached its capacity.

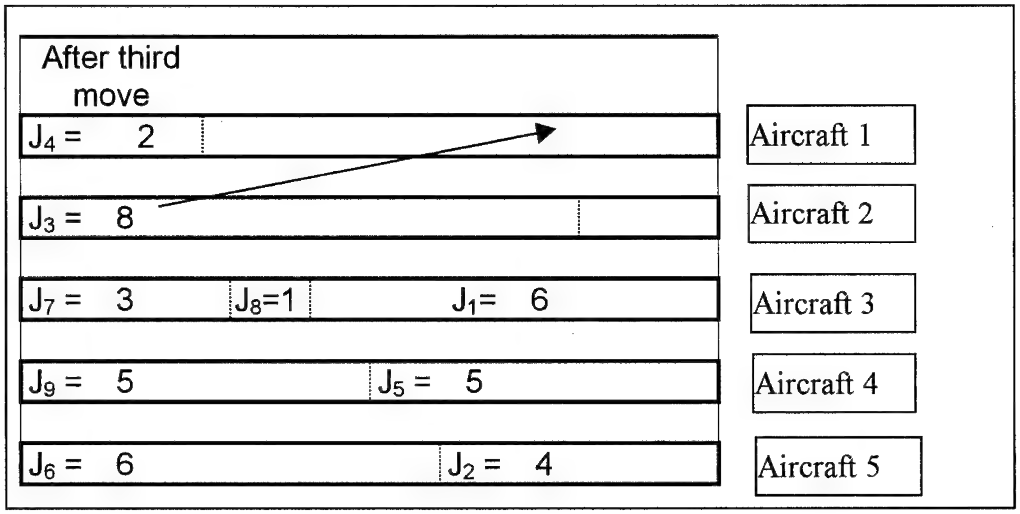


Figure III.4. After third move

Mission 3 is moved from aircraft 2 to aircraft 1. Aircraft 1 has reached its capacity.

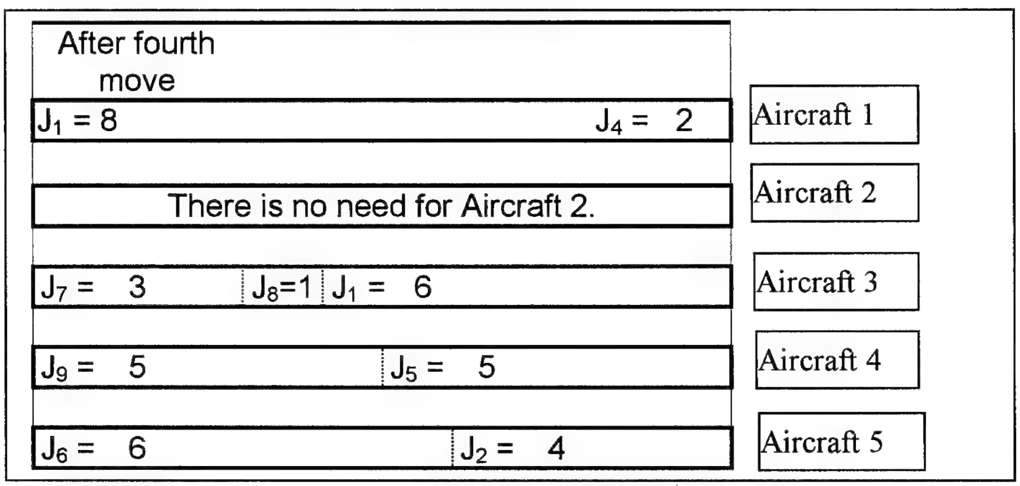


Figure III.5. After the fourth move

After the third move we see that four aircraft are sufficient to process the missions. However, as the problem gets bigger, the time to evaluate and make moves get much longer.

There are several heuristics for the job-machine sequencing problem. In the previous example, we explored the application of a simple heuristic. Another example of a heuristic follows. First, the problem $P_m \,||\, C_{max}$ is considered. This problem is of interest because minimizing the makespan has the effect of balancing the load over the various machines, which is important in practice. We show how the following is related to our problem. The definition of the notation is given below to facilitate understanding the problem.

$\underline{P_m}$: There are m identical machines in parallel. Job j requires a single operation and may be processed on any one of the m machines.

Makespan (C_{max}): The makespan, defined as $\max(C_1, C_2, \ldots\ldots, C_n)$, is equivalent to the completion time of the last job to leave the system. A minimum makespan usually implies a high utilization of the machines.

The *longest processing time first (LPT)* rule assigns the *m* largest jobs to *m* machines. After that, whenever a machine is freed, the largest unscheduled job is put on the machine. This heuristic tries to place the shorter jobs toward the end of the schedule where they can be used for balancing the loads.

Now consider four parallel machines with the capacity of 12 units and nine jobs whose processing times are given in the following table :

Table III-1. Jobs and Processing Times

JOBS	J1	J2	J3	J4	J5	J6	J7	J8	J9
PROCESSING TIMES	7	7	6	6	5	5	4	4	4

The LPT heuristic for $P_m \,||\, C_{max}$ is:

Step 1: Construct an LPT ordering of the jobs.

Step 2: Schedule the jobs in LPT order, each time assigning a job to the machine with the least amount of processing already assigned.

In this example the jobs are already given in the longest processing time order.

C_{max} (LPT) = 15

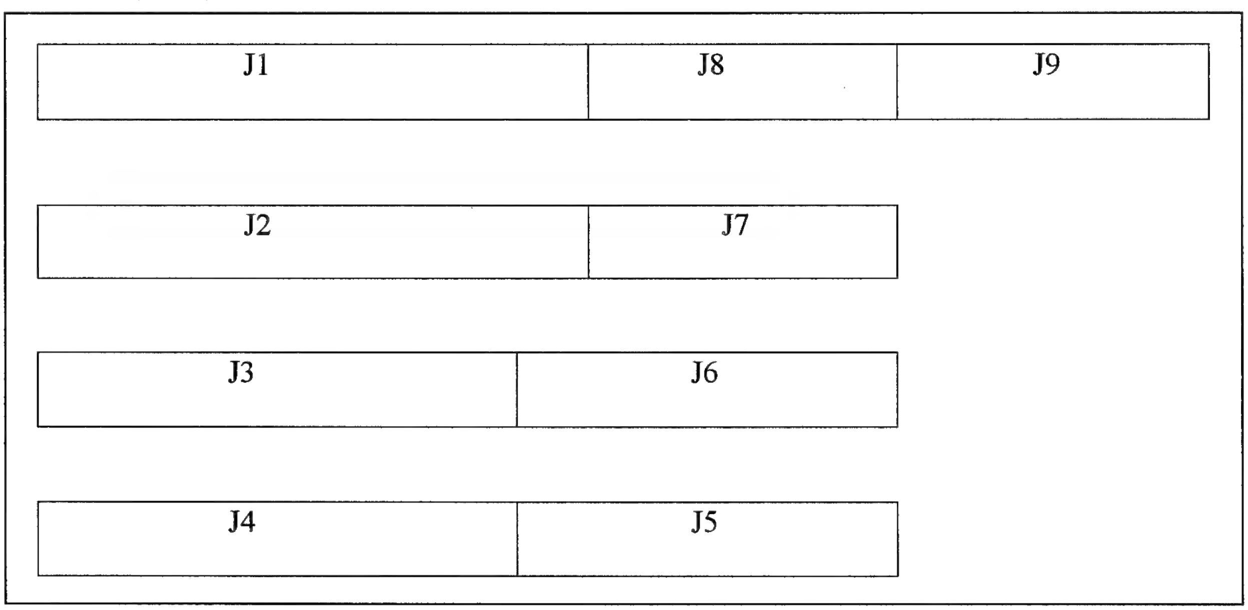


Figure III.6. LPT Solution

We can conclude that we need 5 machines to complete the jobs if the capacity of a machine is 12 units, because with 4 machines, the makespan exceeds the capacity of the machine. The heuristic presented above cannot guarantee an optimal makespan, but it often gives good solutions.

A theorem gives the upper bound of $C_{max}(LPT)/C_{max}(OPT)$ for this heuristic, where $C_{max}(LPT)$ denotes the makespan of the LPT schedule and $C_{max}(OPT)$ denotes the makespan of the (possibly unknown) optimal schedule. This type of worst case analysis is interesting because it gives an indication of how well the heuristic is guaranteed to perform as well as the type of instances for which the heuristic performs badly.

Theorem: For $P_m \,||\, C_{max}$:

$$\frac{C_{\max}(LPT)}{C_{\max}(OPT)} \leq \frac{4}{3} - \frac{1}{3m}$$

In our example above with four machines (m=4) the C_{max} can be:

$$C_{\max}(OPT) \geq \frac{C_{\max}(LPT)}{\frac{4}{3} - \frac{1}{3m}}$$

$C_{max}(OPT) \geq 12$

In this case, we cannot say our solution is optimal since C_{max} can be as low as 12 units. We can prove that C_{max} can be 12 units with the following solution:

$C_{max}(OPT) = 12$ units

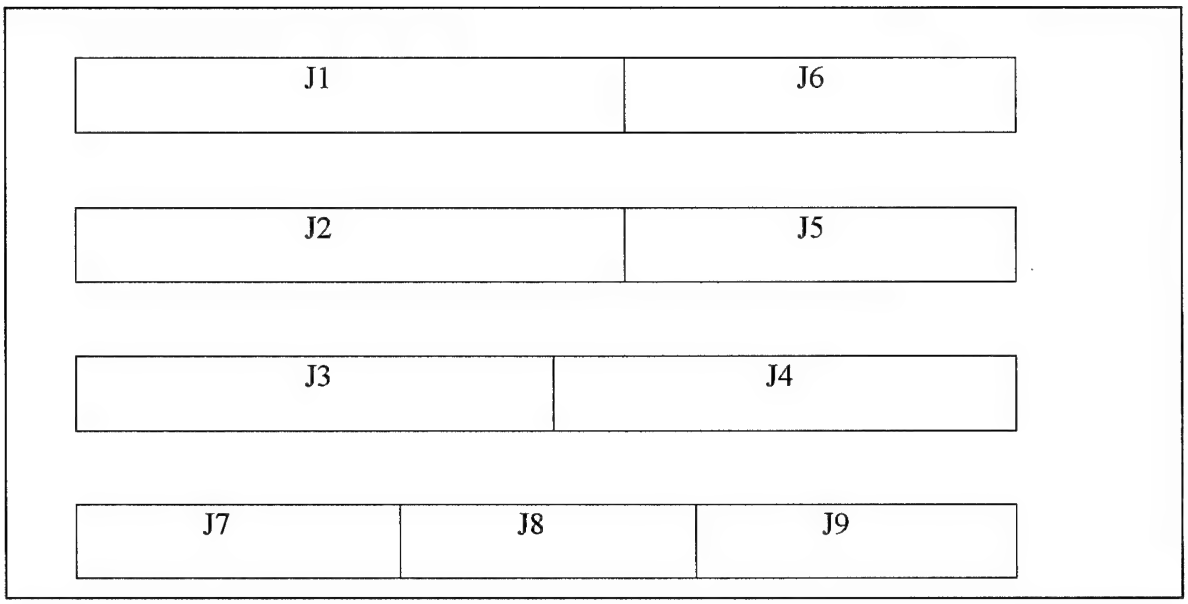


Figure III.7. Optimal Solution

Since the LPT heuristic does not guarantee the optimal solution, we formulated our research problem as an integer program based on the job-parallel machine model.

III.2 THE PROBLEM FORMULATION

The Turkish Air Mobility Command channel cargo system can be viewed as a type of job – shop-scheduling problem. As presented in Chapter II, a job – shop - scheduling problem represents a system as a set of jobs.

Viewing the channel cargo distribution system as a job – machine- scheduling problem, a machine corresponds to an aircraft, and a job refers to a mission to be completed. The following assumptions are made:

- Each job is a single entity.
- No preemption is allowed. Once a job starts on a machine, it will complete processing on that machine.
- No job can be canceled.
- There is only one type of machine.
- There is no randomness: The number of jobs and their processing times are known deterministically.

III.2.1 PARALLEL IDENTICAL PROCESSORS AND INDEPENDENT JOBS

In scheduling, it is often advantageous to use the parallelism in resource structure. We will assume a static case with *m* identical machines available for processing, and a job can be processed by at most one machine at a time. As mentioned

in the assumptions, the TAMC cargo channel system can be viewed as a job shop-scheduling problem where preemption in the system is not allowed.

The integer programming formulation of the problem $P_m \,||\, C_{max}$ can be constructed as follow:

$$x_{ij} = \begin{cases} 1 \text{ if job i is assigned to machine j} \\ 0 \text{ otherwise} \end{cases}$$

$$\text{Min} \qquad C_{max} \qquad (1)$$

$$\text{Subject to} \qquad C_{max} - \sum_{i=1}^{n} p_i\, x_{ij} \geq 0 \qquad 1 \leq j \leq m \qquad (2)$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad 1 \leq i \leq n \qquad (3)$$

$$x_{ij} \geq 0 \qquad \text{and integer} \qquad (4)$$

C_{max} : (Makespan) The completion time of the last job leaving the system.

P_i : Process time of job i

m : The number of the machines

n : The number of the jobs

The formulation contains *(m + n)* constraints in *(nm+1)* variables. Constraints (3) represent special ordered sets of type 1 (SOS 1) which can be exploited. Even if one does not have access to a code that takes direct advantage of SOS 1 constraints, it has been found useful to list them first in the input matrix. SOS 1 constraints are often found in scheduling problems. (Morton and Pentico, 1993: 241-242)

III.2.2 MODELING THE PROBLEM AS AN INTEGER PROGRAMM

The integer programming formulation of the research problem can be formulated similar to the parallel machine-scheduling model. The objective function is to minimize the number of aircraft used. The makespan reflects the weekly flight hours for the aircraft. In this case, the makespan is known but the minimum number of aircraft needed is unknown. After some modification, the following integer-programming formulation can be used to help determine the number of aircraft required. The advantages and weaknesses of this approach are discussed in Chapter IV.

FORMULATION

$$\text{Min} \quad \sum_{j=1}^{m} x_j \tag{5}$$

$$\text{Subject to} \quad C_{max} - \sum_{i=1}^{n} p_i \, x_{ij} \geq 0 \qquad 1 \leq j \leq m \tag{6}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad 1 \leq i \leq n \tag{7}$$

$$n \, x_j - \sum_{i=1}^{n} x_{ij} \geq 0 \qquad 1 \leq j \leq m \tag{8}$$

$$x_{ij} \geq 0 \qquad \text{and binary} \tag{9}$$

$$x_j \geq 0 \qquad \text{and binary} \tag{10}$$

$$x_j = \begin{cases} 1 & \text{if aircraft j is assigned} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if mission i is assigned to aircraft j} \\ 0 & \text{otherwise} \end{cases}$$

m: upper bound on number of aircraft

n: number of missions to be completed

p_i: flight time of mission i

C_{max}: maximum available flying hours on each aircraft.

III.2.3 EXAMPLE PROBLEM

We present the following small example of our problem.

MISSIONS	***TIME (hour)***
J_1	5
J_2	6
J_3	5
J_4	3
J_5	2
J_6	6

We assume the aircraft can be used at most 6 hours in a work time period. The symbol J_i refers to mission i. With m = 6, the problem is solved with the binary restriction relaxed. This tests to see how the solution time improved and if the solution was reasonable, when the binary restriction was relaxed.

```
MIN      X1 + X2 + X3 + X4 + X5 + X6


SUBJECT TO
          2)   5 X11 + 6 X21 + 5 X31 + 3 X41 + 2 X51 + 6 X61 <=   6
          3)   5 X12 + 6 X22 + 5 X32 + 3 X42 + 2 X52 + 6 X62 <=   6
          4)   5 X13 + 6 X23 + 5 X33 + 3 X43 + 2 X53 + 6 X63 <=   6
          5)   5 X14 + 6 X24 + 5 X34 + 3 X44 + 2 X54 + 6 X64 <=   6
          6)   5 X15 + 6 X25 + 5 X35 + 3 X45 + 2 X55 + 6 X65 <=   6
          7)   5 X16 + 6 X26 + 5 X36 + 3 X46 + 2 X56 + 6 X66 <=   6
          8)   X11 + X12 + X13 + X14 + X15 + X16 =    1
          9)   X21 + X22 + X23 + X24 + X25 + X26 =    1
         10)   X31 + X32 + X33 + X34 + X35 + X36 =    1
         11)   X41 + X42 + X43 + X44 + X45 + X46 =    1
         12)   X51 + X52 + X53 + X54 + X55 + X56 =    1
         13)   X61 + X62 + X63 + X64 + X65 + X66 =    1
         14) - 6 X1 + X11 + X21 + X31 + X41 + X51 + X61 <=   0
         15) - 6 X2 + X12 + X22 + X32 + X42 + X52 + X62 <=   0
         16) - 6 X3 + X13 + X23 + X33 + X43 + X53 + X63 <=   0
         17) - 6 X4 + X14 + X24 + X34 + X44 + X54 + X64 <=   0
         18) - 6 X5 + X15 + X25 + X35 + X45 + X55 + X65 <=   0
         19) - 6 X6 + X16 + X26 + X36 + X46 + X56 + X66 <=   0
 END
```

Table III-2. Example Problem Relaxed Solution

	OBJECTIVE FUNCTION VALUE	
1)	1.000000	
VARIABLE	VALUE	REDUCED COST
X1	0.250000	0.000000
X2	0.194444	0.000000
X3	0.055556	0.000000
X4	0.250000	0.000000
X5	0.027778	0.000000
X6	0.222222	0.000000
X11	1.000000	0.000000
X21	0.000000	0.000000
X31	0.000000	0.000000
X41	0.000000	0.000000
X51	0.500000	0.000000
X61	0.000000	0.000000
X12	0.000000	0.000000
X22	0.166667	0.000000
X32	1.000000	0.000000
X42	0.000000	0.000000
X52	0.000000	0.000000
X62	0.000000	0.000000
X13	0.000000	0.000000
X23	0.333333	0.000000
X33	0.000000	0.000000
X43	0.000000	0.000000
X53	0.000000	0.000000
X63	0.000000	0.000000
X14	0.000000	0.000000
X24	0.500000	0.000000
X34	0.000000	0.000000
X44	1.000000	0.000000
X54	0.000000	0.000000
X64	0.000000	0.000000
X15	0.000000	0.000000
X25	0.000000	0.000000
X35	0.000000	0.000000
X45	0.000000	0.000000
X55	0.000000	0.000000
X65	0.166667	0.000000
X16	0.000000	0.000000
X26	0.000000	0.000000
X36	0.000000	0.000000
X46	0.000000	0.000000
X56	0.500000	0.000000
X66	0.833333	0.000000

As we see in the solution, many decision variables have noninteger values. Although this example solved very quickly, we have to restrict the variables to binary in order to get a feasible solution. The solution with binary variables is shown below:

Table III-3. Example Problem Integer Solution

OBJECTIVE FUNCTION VALUE		
1)	5.000000	
VARIABLE	VALUE	REDUCED COST
X1	1.000000	1.000000
X2	0.000000	1.000000
X3	1.000000	1.000000
X4	1.000000	1.000000
X5	1.000000	1.000000
X6	1.000000	1.000000
X11	1.000000	0.000000
X21	0.000000	0.000000
X31	0.000000	0.000000
X41	0.000000	0.000000
X51	0.000000	0.000000
X61	0.000000	0.000000
X12	0.000000	0.000000
X22	0.000000	0.000000
X32	0.000000	0.000000
X42	0.000000	0.000000
X52	0.000000	0.000000
X62	0.000000	0.000000

X13	0.000000	0.000000
X23	0.000000	0.000000
X33	1.000000	0.000000
X43	0.000000	0.000000
X53	0.000000	0.000000
X63	0.000000	0.000000
X14	0.000000	0.000000
X24	0.000000	0.000000
X34	0.000000	0.000000
X44	1.000000	0.000000
X54	1.000000	0.000000
X64	0.000000	0.000000
X15	0.000000	0.000000
X25	1.000000	0.000000
X35	0.000000	0.000000
X45	0.000000	0.000000
X55	0.000000	0.000000
X65	0.000000	0.000000
X16	0.000000	0.000000
X26	0.000000	0.000000
X36	0.000000	0.000000
X46	0.000000	0.000000
X56	0.000000	0.000000
X66	1.000000	0.000000

Increasing the number of aircraft, m, does not change the optimal solution. Another solution with m chosen as 7 gives the same solution (see Appendix C). But if

we choose m less than or equal to 4, then the solution set becomes infeasible, because the missions can not be covered with the aircraft.

We experienced a small increase in solution time for this example when binary restriction on the variables were enforced.

III.3 MODEL RESTRICTIONS AND THE SOLUTIO N

In the model, there are some restrictions that need to be explained. These restrictions affect the model and its solution. Firstly, we must define our variables as binary, although the solution time gets longer.

The other limitation would be choosing the proper m for the possible maximum aircraft number. If we choose m too small, there is no feasible solution. On the other hand, if we choose m too large, the solve time will take longer. Therefore we need to be careful in choosing m.

III.4 SOLVER

Hyper LINGO is used to solve the model. LINGO is a mathematical modeling language. Unlike conventional programming languages, such as Basic or C, LINGO is nonprocedural. That is, when you specify a model for LINGO to solve, you only tell it

what you want, not how it should find the solution. In this sense, LINGO is known as a specification language. You tell it what you want, and it does the rest.

LINGO solves formulations with up to 4000 constraints and 800 integer variables. LINGO performs integer programming by using the branch-and-bound and cut-method. (Lindo System Inc., 1996: 171-172) LINGO allows you to code the constraint set so this facilitates the formulation of the problem. It also provides a separate solution sheet. The code and extensive formulation of the main problem can be found in Appendix A and Appendix B.

CHAPTER IV

IV. ANALYSIS AND RESULTS

In this chapter, the methodology is tested. The steps from obtaining of data to getting results are discussed. As discussed in Chapter III, the problem was formulated as an integer program. Section IV.1 is about the formation of the data. The data is statistically analyzed in Section IV.2. In Section IV.3, the solution process and the results are described.

IV.1 THE FORMATION OF THE DATA

I examined the flight schedule in order to get insight about the planning. I have seen that the schedules are made according to missions. I collected my model inputs from a twenty-two week schedule (peacetime) using statistical analysis. I came up with the mission chart(see Appendix D) over the twenty-two-week-schedule for C –130 Hercules aircraft. The purpose of examining the schedule is to come up with a set of requirements. There are three different major mission types.

The three major groups can be described as follows:

- Training
    - Low-altitude cruise, high altitude cruise, search and rescue
- Passenger transport

- Daily tours from origin to destination with return to origin

- Routine resupply

  - Flights from origin to destination and return to origin base

For security reasons, the missions are described with symbols. The schedules of some missions are flexible. For instance, the schedulers can postpone any training during the following week. Thus, the number of training missions flown in a week is subject to change.

On the other hand, there are missions such as routine resupply and passenger transport that have specified time intervals. They have to be flown at that assigned time interval. The formation of a mission chart is shown on the next pages along with a statistical analysis.

IV.2 THE STATICTIAL ANALYSIS OF THE DATA

In order to form a general mission chart, descriptive statistics are applied. The missions are examined individually along the twenty-two week schedule. First, the training missions are examined. There are eight different types of training missions flown in the twenty-two week schedule. The Ti symbol refers to training mission type i. The statistics obtained from this twenty-two week schedule are shown on the next pages. In Table IV-1, the number of times each mission is flown over the twenty-two week schedule is shown.

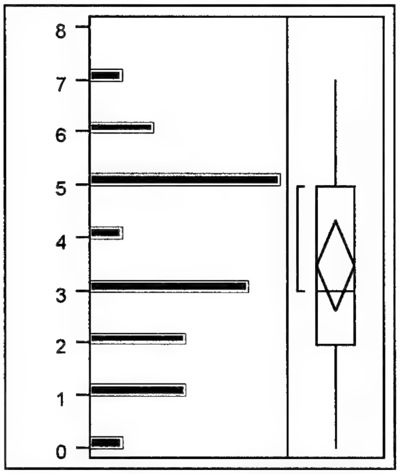
Table IV-1. Data From The 22 – Weekly Schedule

WEEK	T1	T2	T3	T4	T5	T6	T7	T8
1	7	7	10	3	4	2	4	2
2	5	7	9	2	4	6	5	8
3	6	5	6	2	4	6	5	7
4	2	7	6	2	3	5	4	6
5	5	7	7	2	4	6	5	9
6	3	6	7	2	3	5	4	7
7	5	6	5	2	3	5	4	6
8	3	6	8	2	0	6	5	8
9	4	6	7	2	3	5	4	5
10	5	6	4	2	3	5	1	6
11	2	7	7	2	2	4	3	4
12	5	7	5	3	4	6	5	8
13	3	7	5	1	3	5	4	5
14	0	7	7	2	1	3	2	1
15	2	6	6	2	5	7	6	10
16	1	6	10	1	3	5	4	6
17	3	7	8	2	4	6	5	9
18	5	7	5	2	4	6	5	9
19	6	6	5	2	2	4	3	4
20	1	6	7	2	4	6	5	8
21	1	6	4	2	3	5	4	5
22	3	7	6	2	4	6	5	9

Each training mission type is examined individually with descriptive statistics. The following tables show the results of these statistics. These statistics give

information about descriptive statistics of data such as mean, standard error and quantiles.

Table IV-2. Descriptive Statistics for Mission T1



Stem	Leaf	Count
7	0	1
6	0 0	2
5	0 0 0 0 0 0	6
4	0	1
3	0 0 0 0 0	5
2	0 0 0	3
1	0 0 0	3
0	0	1

Quantiles

Maximum	100.0%	7.0000
	99.5%	7.0000
	97.5%	7.0000
	90.0%	6.0000
quartile	75.0%	5.0000
median	50.0%	3.0000
quartile	25.0%	2.0000
	10.0%	1.0000
	2.5%	0.0000
	0.5%	0.0000
minimum	0.0%	0.0000

Moments

Mean	3.50000
Std Dev	1.92106
Std Error Mean	0.40957
Upper 95% Mean	4.35174
Lower 95% Mean	2.64826
N	22.00000
Sum Weights	22.00000
Sum	77.00000
Variance	3.69048
Skewness	-0.04433
Kurtosis	-0.96111
CV	54.88746

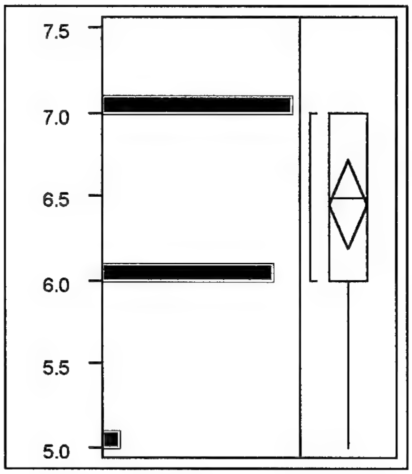
Test for Normality

Shapiro-Wilk W Test

W	Prob<W
0.948110	0.2878

As we see from the statistics, mission T1 has a mean of 3.5. T1 mission has the time – length of 140 minutes. This also includes ground time operations.

Table IV-3. Descriptive Statistics for Mission T2



Stem	Leaf	Count
70	00000000000	11
68		
66		
64		
62		
60	0000000000	10
58		
56		
54		
52		
50	0	1

Multiply Stem.Leaf by 10^-1

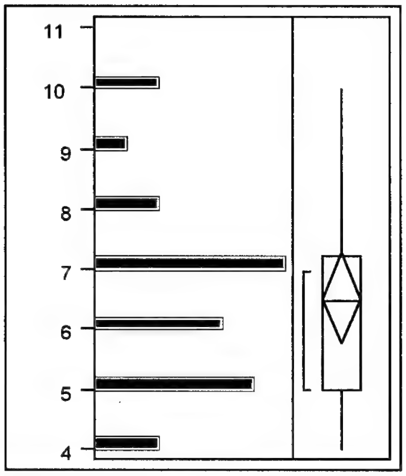
Quantiles		
maximum	100.0%	7.0000
	99.5%	7.0000
	97.5%	7.0000
	90.0%	7.0000
quartile	75.0%	7.0000
median	50.0%	6.5000
quartile	25.0%	6.0000
	10.0%	6.0000
	2.5%	5.0000
	0.5%	5.0000
minimum	0.0%	5.0000

Moments	
Mean	6.45455
Std Dev	0.59580
Std Error Mean	0.12703
Upper 95% Mean	6.71871
Lower 95% Mean	6.19038
N	22.00000
Sum Weights	22.00000
Sum	142.00000
Variance	0.35498
Skewness	-0.55265
Kurtosis	-0.52436
CV	9.23071

Test for Normality	
Shapiro-Wilk W Test	
W	Prob<W
0.733345	<.0001

T2 has a mean of 6.45, and T2 is a 210-minute – mission.

Table IV-4. Descriptive Statistics for Mission T3



Stem	Leaf	Count
10	0 0	2
9	0	1
8	0 0	2
7	0 0 0 0 0 0	6
6	0 0 0 0	4
5	0 0 0 0 0	5
4	0 0	2

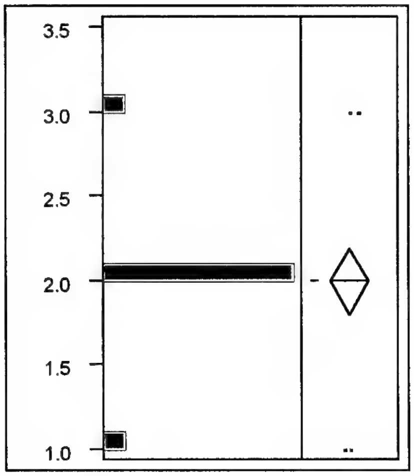
Quantiles		
Maximum	100.0%	10.000
	99.5%	10.000
	97.5%	10.000
	90.0%	9.700
quartile	75.0%	7.250
median	50.0%	6.500
quartile	25.0%	5.000
	10.0%	4.300
	2.5%	4.000
	0.5%	4.000
minimum	0.0%	4.000

Moments	
Mean	6.54545
Std Dev	1.71067
Std Error Mean	0.36472
Upper 95% Mean	7.30392
Lower 95% Mean	5.78699
N	22.00000
Sum Weights	22.00000
Sum	144.00000
Variance	2.92641
Skewness	0.54271
Kurtosis	-0.20017
CV	26.13530

Test for Normality	
Shapiro-Wilk W Test	
W	Prob<W
0.930397	0.1232

The T3 mean is 6.5ansd the time span for this mission is 195 minutes.

Table IV-5. Descriptive Statistics for Mission T4



Stem	Leaf	Count
30	0 0	2
28		
26		
24		
22		
20	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	18
18		
16		
14		
12		
10	0 0	2

Multiply Stem.Leaf by 10^-1

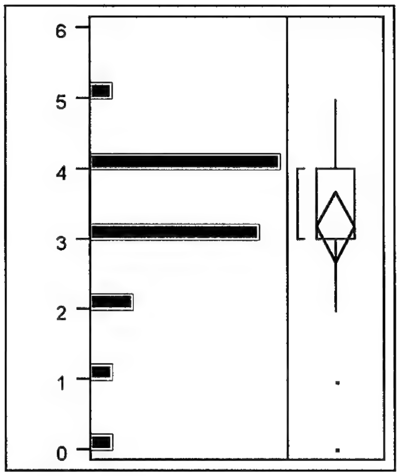
Quantiles		
maximum	100.0%	3.0000
	99.5%	3.0000
	97.5%	3.0000
	90.0%	2.7000
quartile	75.0%	2.0000
median	50.0%	2.0000
quartile	25.0%	2.0000
	10.0%	1.3000
	2.5%	1.0000
	0.5%	1.0000
minimum	0.0%	1.0000

Moments	
Mean	2.00000
Std Dev	0.43644
Std Error Mean	0.09305
Upper 95% Mean	2.19350
Lower 95% Mean	1.80650
N	22.00000
Sum Weights	22.00000
Sum	44.00000
Variance	0.19048
Skewness	0.00000
Kurtosis	3.50921
CV	21.82179

Test for Normality	
Shapiro-Wilk W Test	
W	Prob<W
0.600030	<.0001

The mission T4 has a mean of 1.9 and duration of 105- minute–period.

Table IV-6. Descriptive Statistics for Mission T5



Stem	Leaf	Count
5	0	1
4	000000000	9
3	00000000	8
2	00	2
1	0	1
0	0	1

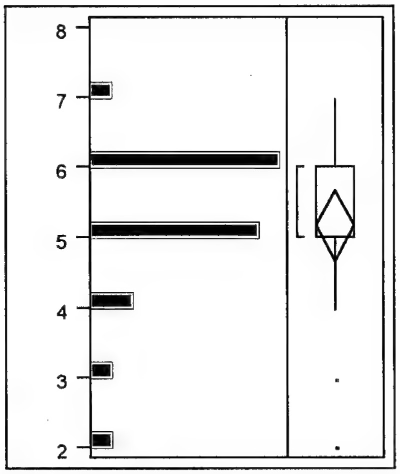
Quantiles		
Maximum	100.0%	5.0000
	99.5%	5.0000
	97.5%	5.0000
	90.0%	4.0000
quartile	75.0%	4.0000
median	50.0%	3.0000
quartile	25.0%	3.0000
	10.0%	1.3000
	2.5%	0.0000
	0.5%	0.0000
minimum	0.0%	0.0000

Moments	
Mean	3.18182
Std Dev	1.13961
Std Error Mean	0.24296
Upper 95% Mean	3.68709
Lower 95% Mean	2.67655
N	22.00000
Sum Weights	22.00000
Sum	70.00000
Variance	1.29870
Skewness	-1.23902
Kurtosis	1.93311
CV	35.81618

Test for Normality	
Shapiro-Wilk W Test	
W	Prob<W
0.846351	0.0022

The mission T5 has the time span of 160 minutes.

Table IV-7. Descriptive Statistics for Mission T6



Stem	Leaf	Count
7	0	1
6	000000000	9
5	00000000	8
4	00	2
3	0	1
2	0	1

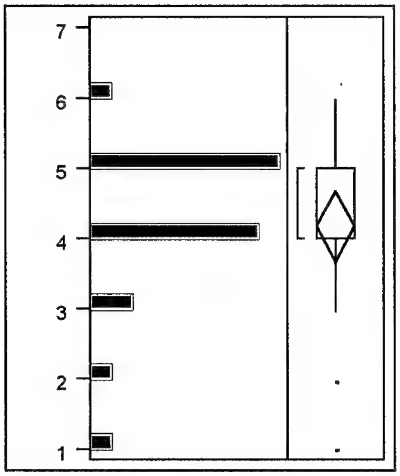
Quantiles		
maximum	100.0%	7.0000
	99.5%	7.0000
	97.5%	7.0000
	90.0%	6.0000
quartile	75.0%	6.0000
median	50.0%	5.0000
quartile	25.0%	5.0000
	10.0%	3.3000
	2.5%	2.0000
	0.5%	2.0000
minimum	0.0%	2.0000

Moments	
Mean	5.18182
Std Dev	1.13961
Std Error Mean	0.24296
Upper 95% Mean	5.68709
Lower 95% Mean	4.67655
N	22.00000
Sum Weights	22.00000
Sum	114.00000
Variance	1.29870
Skewness	-1.23902
Kurtosis	1.93311
CV	21.99239

Test for Normality	
Shapiro-Wilk W Test	
W	Prob<W
0.846351	0.0022

The time period of the mission T6 is 255 minutes.

Table IV-8. Descriptive Statistics for Mission T7



Stem	Leaf	Count
6	0	1
5	0 0 0 0 0 0 0 0 0	9
4	0 0 0 0 0 0 0 0	8
3	0 0	2
2	0	1
1	0	1
0		

Quantiles

Maximum	100.0%	6.0000
	99.5%	6.0000
	97.5%	6.0000
	90.0%	5.0000
quartile	75.0%	5.0000
median	50.0%	4.0000
quartile	25.0%	4.0000
	10.0%	2.3000
	2.5%	1.0000
	0.5%	1.0000
minimum	0.0%	1.0000

Moments

Mean	4.18182
Std Dev	1.13961
Std Error Mean	0.24296
Upper 95% Mean	4.68709
Lower 95% Mean	3.67655
N	22.00000
Sum Weights	22.00000
Sum	92.00000
Variance	1.29870
Skewness	-1.23902
Kurtosis	1.93311
CV	27.25144

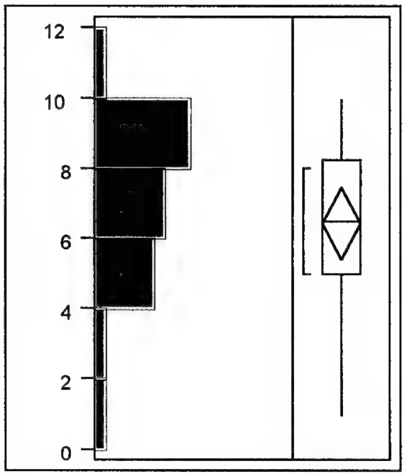
Test for Normality

Shapiro-Wilk W Test

W	Prob<W
0.846351	0.0022

Mission T7 has a 225 minute time period.

Table IV-9. Descriptive Statistics for Mission T8



Stem	Leaf	Count
10	0	1
9	0 0 0 0	4
8	0 0 0 0	4
7	0 0	2
6	0 0 0 0	4
5	0 0 0	3
4	0 0	2
3		
2	0	1
1	0	1
0		

Quantiles		
Maximum	100.0%	10.000
	99.5%	10.000
	97.5%	10.000
	90.0%	9.000
quartile	75.0%	8.250
median	50.0%	6.500
quartile	25.0%	5.000
	10.0%	2.600
	2.5%	1.000
	0.5%	1.000
minimum	0.0%	1.000

Moments	
Mean	6.45455
Std Dev	2.36497
Std Error Mean	0.50421
Upper 95% Mean	7.50311
Lower 95% Mean	5.40599
N	22.00000
Sum Weights	22.00000
Sum	142.00000
Variance	5.59307
Skewness	-0.65036
Kurtosis	-0.00724
CV	36.64035

Test for Normality	
Shapiro-Wilk W Test	
W	Prob<W
0.942751	0.2240

Mission T8 has a 160 minute time span.

The time periods for the missions also include the ground – time – operation. With respect to the above descriptive statistics, I developed the following general weekly training schedule by rounding up to the next highest value of the mission number's mean.

Table IV-10. The General Mission Chart for Training Missions

MISSION	T1	T2	T3	T4	T5	T6	T7	T8
# MISSIONS IN A WEEK	4	7	7	2	4	6	5	7
TIME(HOURS) PER MISSION	2.333	3.5	3.25	1.75	2.66	4.25	3.75	2.66
TIME(MIN.) PER MISSION	140	210	195	105	160	255	225	160

As mentioned in the beginning of the chapter, another category of missions is based on passenger transportation. The schedule for this mission is standard and operates on the room – available rule for the military personnel. On the other hand, it also carries cargo with the same procedure. For the security reasons the schedule is modified in a reasonable manner. The schedule is shown in the following table.

Table IV-11. The Mission Chart Based on Passenger Transportation

FLIGHT MISSION	MISSION TIME (HOURS)	MISSION TIME (MIN.)	START TIME	DUE TIME
R1	8.25	495	MONDAY 8^{00}	MONDAY 16^{15}
R2	13.5	810	TUESDAY 7^{30}	TUESDAY 21^{00}
R3	12.25	735	WEDNESDAY 7^{00}	WEDNESDAY 19^{15}
R4	7.25	435	THURSDAY 7^{45}	THURSDAY 15^{00}
R5	14.25	855	FRIDAY 7^{30}	FRIDAY 21^{45}
R6	9.25	555	---	---
R7	8.25	495	THURSDAY 8^{30}	THURSDAY 15^{45}

The last group of missions is cargo missions. The missions are planned from the origin base to destination base and back to the origin base. Figure VI-1 shows the destination bases and great circle distances from the origin bases to the destination base. Missions start and end at the origin base. As we know, the great circle distance can be used only if we fly directly to destination. In normal cases, we have to fly specified air routes, which are not directly from the origin base to the destination base. The service time and the loading/unloading time are included in the flight time computation. The cargo mission chart is shown in Figure IV-1.

Mission ID	Destination ICAO ID	Great Circle Distance (NM)	Flight Time (Origin-Destination-Origin)		Origin Information			Destination Information		
			Hours	Minutes	Name	Latitude	Longitude	Name	Latitude	Longitude
C1	LT1X	379	5.5	330	ERKILET	038 46.0 N	035 33.0 E	CORLU	041 08.0 N	027 54.0 E
C2	LTBA	338	4.3	258	ERKILET	038 46.0 N	035 33.0 E	ATATURK	040 58.0 N	028 48.0 E
C3	LTBG	362	4.9	294	ERKILET	038 46.0 N	035 33.0 E	BANDIRMA	040 19.0 N	027 59.0 E
C4	LTBF	358	4.8	288	ERKILET	038 46.0 N	035 33.0 E	BALIKESIR	039 37.0 N	027 55.0 E
C5	LTBL	400	5.7	342	ERKILET	038 46.0 N	035 33.0 E	CIGLI	038 31.0 N	027 01.0 E
C6	LTBR	291	3.8	228	ERKILET	038 46.0 N	035 33.0 E	YENISEHIR	040 15.0 N	029 34.0 E
C7	LTCC	225	3.4	204	ERKILET	038 46.0 N	035 33.0 E	DIYARBAKIR	037 53.0 N	040 12.0 E
C8	LTCI	365	5.1	306	ERKILET	038 46.0 N	035 33.0 E	VAN	038 28.0 N	043 20.0 E

Figure IV-1. Routine Resupply Missions

After I finished the formation of the mission schedules, I assigned a variable for each mission in order to formulate my problem. (see Appendix D.) The model has $(n \times m) + m$ binary variables and $(2m + n)$ constraints. Here n represents the number missions and m represents an upper bound on the number of aircraft.

IV.3 SOLUTION PROCESS

We solved two versions of the problem, varying the restriction on the decision variables in each version. Before examining these two versions, we relaxed all binary restriction on variables. This the solution provided no insight.

- Mixed integer

    The goal of this stage is to get a lower bound on the number of aircraft needed. Therefore aircraft associated variables (x_j) are defined as binary and the assignment variables (x_{ij}) as noninteger. (see Appendix E for extensive solution of mixed integer). The formulation generated in this stage is the same as formulating the problem with preemption as in job – shop modeling. It allocates a mission to more than one aircraft. That is an aircraft can do a fraction of a mission. We found the lower bound on the number of aircraft was nine.

- Integer programming formulation

As a final stage, I solved the problem as an integer program where all variables are binary. In order to improve the solution time, some modifications were made on the formulation. A modification is made by changing the equality constraint (7) to inequality constraints, because of the solver structure. (see Appendix F for

modification). The modification provided an optimal solution and improved the solution time. The solve time dropped from thirty hours to fourteen hours.

The problem was solved in three ways. First, the mean number of training missions was rounded up to the next highest value. Second, the number of training missions was rounded down to next lowest value. Third, the number of training missions of each type was set at its highest possible value. The following graphs show the solution with the aircraft utilization for the first two case.

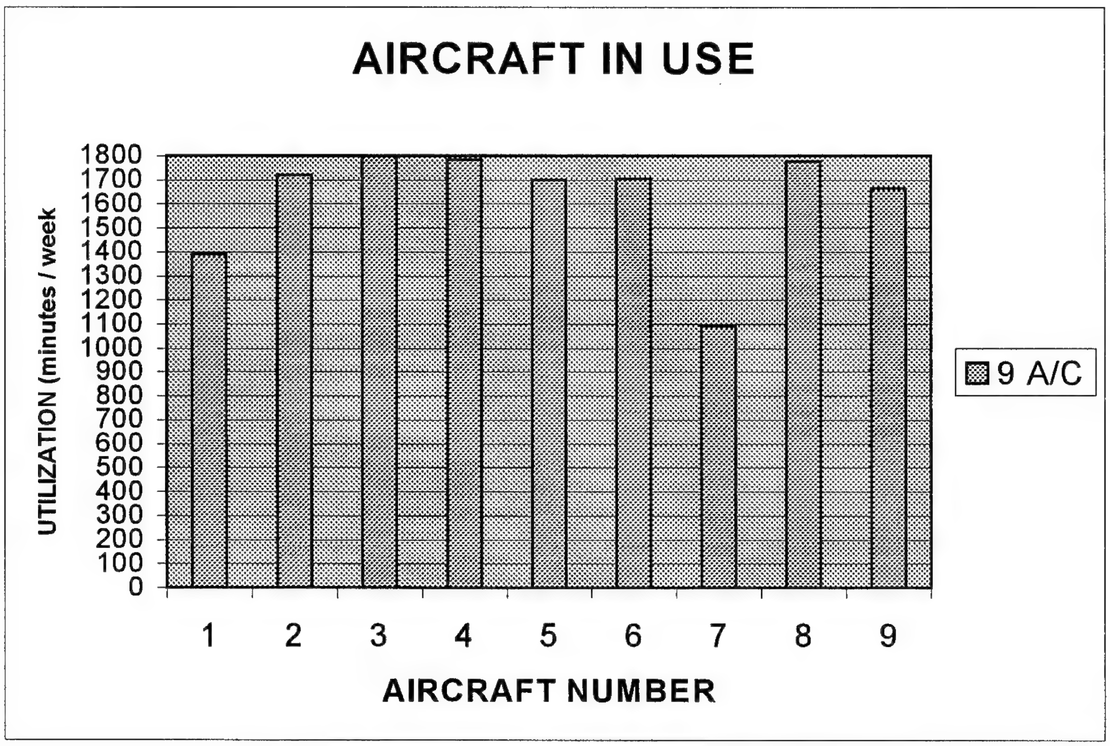


Figure IV-2. Aircaraft Utilizations (9 A/C)

Figure IV-2 shows the solution when the number of training missions was rounded up to the next highest value. For this case, the total number of missions flown in a week is 57, and the average aircraft utilization is 90.3 % or 27.09 hours per week.

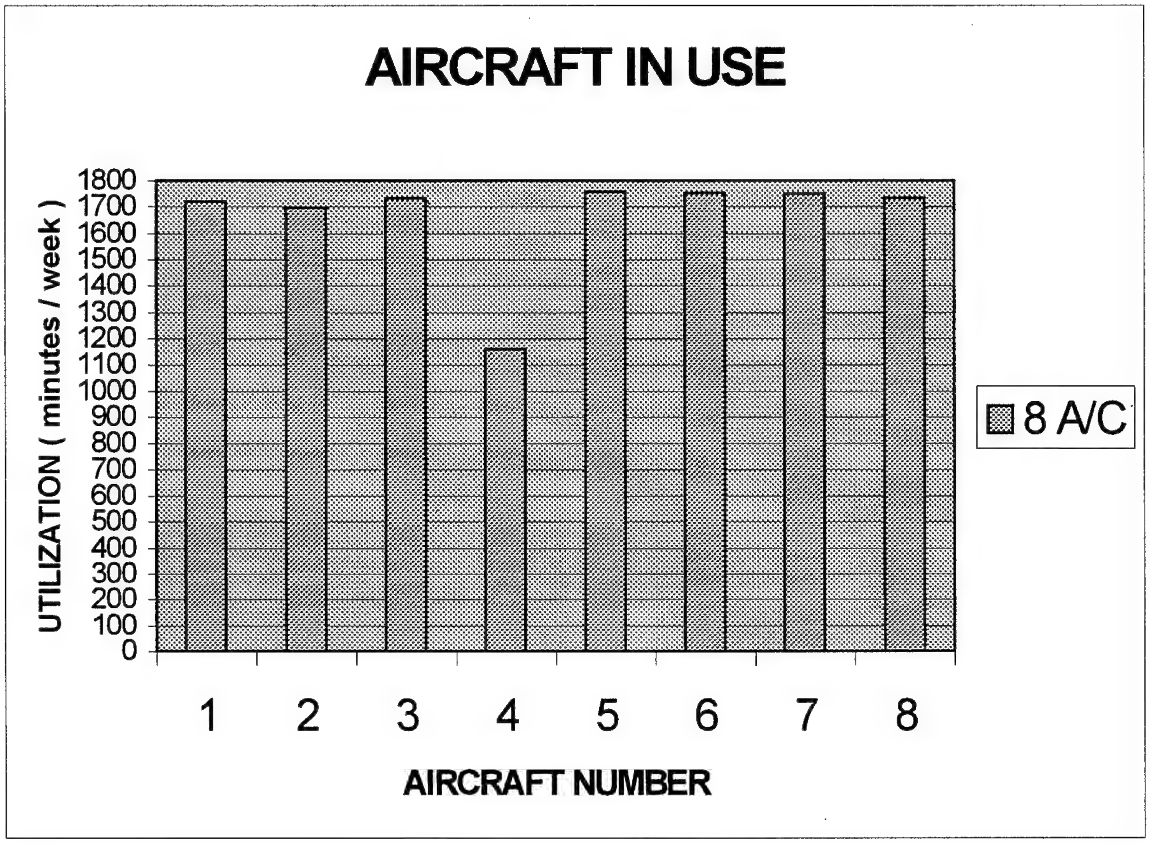


Figure IV-3. Aircraft Utilization (8 A/C)

Figure IV-3 shows the solution when the number of training mission was rounded down to the next lowest value. Here, the total number of missions was 50, and the average aircraft utilization is 92.4 % or 27.72 hours per week.

If we take the lower values of the means we decrease the number of aircraft needed by one. Since we reduced the number of missions by rounding down the means to the next lowest values, the number of aircraft needed is decreased by one. But we would not be able to cover all missions. Therefore we need to take upper values in order to guarantee coverage of all the missions. (See Appendix G for extensive Lingo Solution).

Lastly, we considered the worst case and took the maximum number of training missions. We call that the worst case because the requirement can not be larger than this number of missions. Figure IV-4 shows the solution when the maximum number of the missions was taken. In this case, the total number of missions was 70, and the average aircraft utilization is 93.8 % or 28.13 hours per week. (See Appendix G for extensive Lingo Solution).

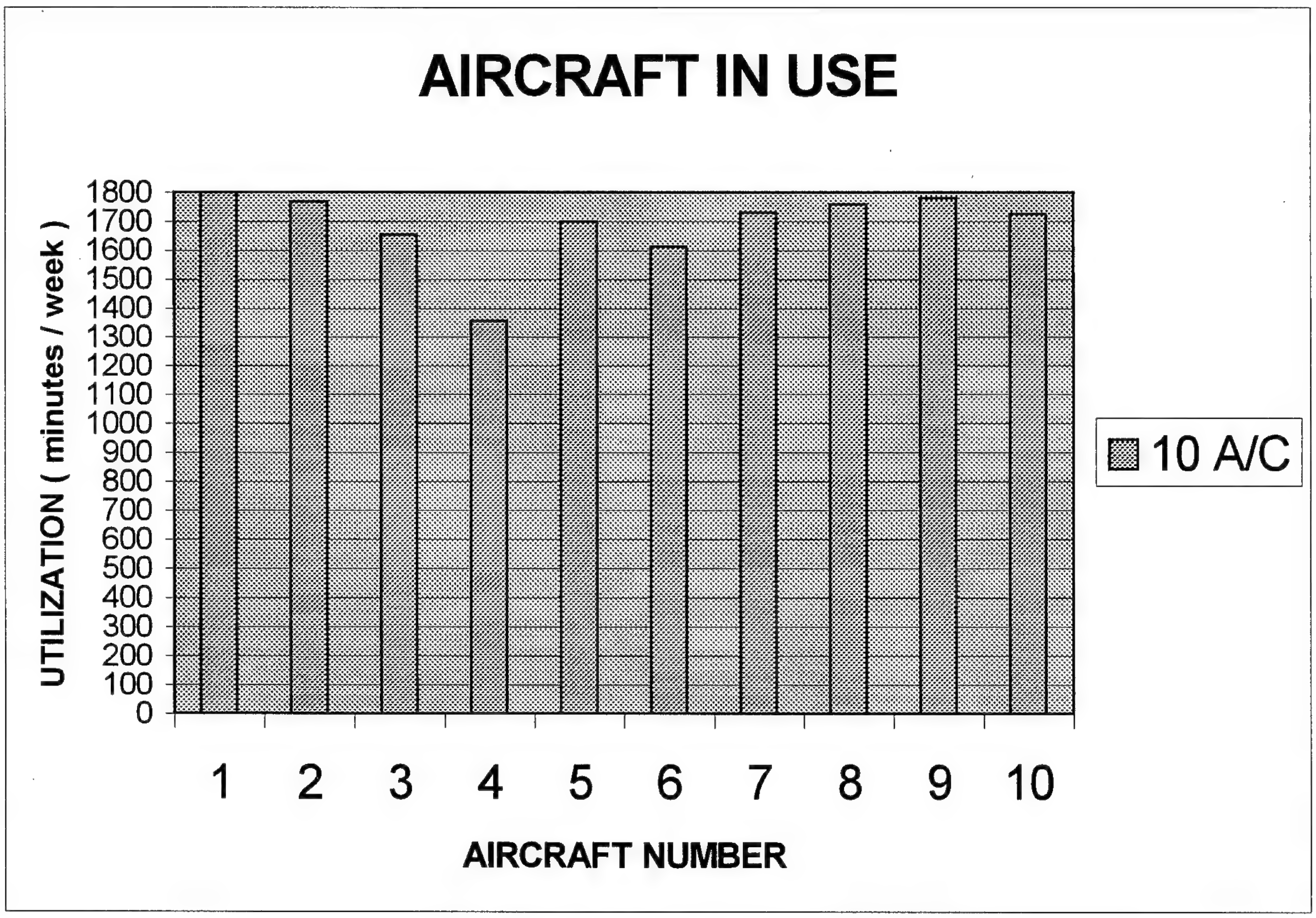


Figure IV-4. Aircraft Utilization (10 A/C)

CHAPTER V

V. CONCLUSIONS AND RECOMMENDATION

This chapter provides a summary of the research and presents ideas for future research involving the use of detailed information to improve the model.

V.1 CONCLUSIONS

This research has established and tested a model for obtaining the information necessary to identify the number of aircraft needed to cover certain missions on a weekly basis. The goal of the IP formulation is to minimize the number of aircraft needed for the missions.

The formulation was generated based on the job – machine – scheduling model. The main formulation was modified because of the computer solver restrictions. The modification is made by changing the special ordered set constraints. Although we increased the number of constraints, it provided a better solution time.

By utilizing the information like that provided by this research, determination can be made on how many aircraft are needed for covering certain missions.

V.2 RECOMMENDATION

Although integer-programming formulation provides an optimal solution, it takes a long time to solve the problem. If the time is limited to solve a problem, it might not be convenient to solve the problem with integer programming.

As mentioned in Chapter III, the problem can be solved with other methods as well. Future research might use another solution method to solve the problem. Then the solution can be compared with the current solution.

Future research might focus on expanding the formulation by adding constraints to include more restrictions on the missions. Additional constraints, which might be added to this model would involve each base's operational capacity and hours of operation. Airbases are assumed to have infinite mission handling capacity, so nothing prevents every aircraft in the system from landing at the same airbase at same time. Also, there are no limits as to the time of day for any activity. The time of day may be an important factor in handling the missions although it did not seem an important factor in this research.

Finally future research could investigate better ways to incorporate these concerns into the formulation, perhaps by modeling the formulation with time windows. It might be considered even other methods to solve the problem such as simulation as well as the methods mentioned in Chapter III. Furthermore, it is wiser to know the infrastructure of the solver, because it might not suit your problem perfectly.

APPENDIX A. THE LINGO CODE

THE ORIGINAL CODE

```
MODEL:

! ASSIGNING MISSION TO AIRCRAFT Problem;

SETS:

   AIRCRAFT / 1..9/      : AIRCRAFTNUMBER ;

   MISSION / 1..57/    : MISSIONLENGTH;

   MISAIR( MISSION, AIRCRAFT) : ASSIGN;

ENDSETS


! The objective;

 [OBJ] MIN = @SUM(AIRCRAFT:AIRCRAFTNUMBER);


! The MISSION-AIRCRAFT ASSIGNING constraints;

  @FOR( MISSION( I):

    @SUM( AIRCRAFT( J): ASSIGN( I, J)) = 1 );



! The AIRCRAFT PURCHASE  constraints;

   @FOR( AIRCRAFT( J):
```

```
    @SUM( MISSION( I): ASSIGN( I, J)) <= 57*AIRCRAFTNUMBER(J));


! The AIRCRAFT CAPACITY  constraints;

  @FOR( AIRCRAFT( J):

    @SUM( MISSION( I): ASSIGN( I, J) * MISSIONLENGTH( I) ) <= 1800);


!DEFINE VARIABLE;

 @FOR( AIRCRAFT: @BIN( AIRCRAFTNUMBER));

 @FOR( MISAIR: @BIN( ASSIGN));


  DATA :

    MISSIONLENGTH = @FILE ( MLDATA.LDT) ;

  ENDDATA


END
```

THE MODIFIED CODE

The special ordered set constraints are modified as follows:

```
! The MISSION-AIRCRAFT ASSIGNING constraints;
 @FOR( MISSION( I):
   @SUM( AIRCRAFT( J): ASSIGN( I, J)) >= .93);

! The MISSION-AIRCRAFT ASSIGNING constraints;
 @FOR( MISSION( I):
   @SUM( AIRCRAFT( J): ASSIGN( I, J)) <= 1.07);
```

APPENDIX B. THE ALGEBRIC FORMULATION

```
Rows=      76 Vars=     522 No. integer vars=     522   ( all are linear)


MIN      AIRCRAFTNUMBER( 1) + AIRCRAFTNUMBER( 2) + AIRCRAFTNUMBER( 3)
     + AIRCRAFTNUMBER( 4) + AIRCRAFTNUMBER( 5) + AIRCRAFTNUMBER( 6)
     + AIRCRAFTNUMBER( 7) + AIRCRAFTNUMBER( 8) + AIRCRAFTNUMBER( 9)
SUBJECT TO
2]   ASSIGN( 1, 1) + ASSIGN( 1, 2) + ASSIGN( 1, 3) + ASSIGN( 1, 4)
     + ASSIGN( 1, 5) + ASSIGN( 1, 6) + ASSIGN( 1, 7) + ASSIGN( 1, 8)
     + ASSIGN( 1, 9) =     1
3]   ASSIGN( 2, 1) + ASSIGN( 2, 2) + ASSIGN( 2, 3) + ASSIGN( 2, 4)
     + ASSIGN( 2, 5) + ASSIGN( 2, 6) + ASSIGN( 2, 7) + ASSIGN( 2, 8)
     + ASSIGN( 2, 9) =     1
4]   ASSIGN( 3, 1) + ASSIGN( 3, 2) + ASSIGN( 3, 3) + ASSIGN( 3, 4)
     + ASSIGN( 3, 5) + ASSIGN( 3, 6) + ASSIGN( 3, 7) + ASSIGN( 3, 8)
     + ASSIGN( 3, 9) =     1
5]   ASSIGN( 4, 1) + ASSIGN( 4, 2) + ASSIGN( 4, 3) + ASSIGN( 4, 4)
     + ASSIGN( 4, 5) + ASSIGN( 4, 6) + ASSIGN( 4, 7) + ASSIGN( 4, 8)
     + ASSIGN( 4, 9) =     1
6]   ASSIGN( 5, 1) + ASSIGN( 5, 2) + ASSIGN( 5, 3) + ASSIGN( 5, 4)
     + ASSIGN( 5, 5) + ASSIGN( 5, 6) + ASSIGN( 5, 7) + ASSIGN( 5, 8)
     + ASSIGN( 5, 9) =     1
7]   ASSIGN( 6, 1) + ASSIGN( 6, 2) + ASSIGN( 6, 3) + ASSIGN( 6, 4)
     + ASSIGN( 6, 5) + ASSIGN( 6, 6) + ASSIGN( 6, 7) + ASSIGN( 6, 8)
     + ASSIGN( 6, 9) =     1
8]   ASSIGN( 7, 1) + ASSIGN( 7, 2) + ASSIGN( 7, 3) + ASSIGN( 7, 4)
     + ASSIGN( 7, 5) + ASSIGN( 7, 6) + ASSIGN( 7, 7) + ASSIGN( 7, 8)
     + ASSIGN( 7, 9) =     1
9]   ASSIGN( 8, 1) + ASSIGN( 8, 2) + ASSIGN( 8, 3) + ASSIGN( 8, 4)
     + ASSIGN( 8, 5) + ASSIGN( 8, 6) + ASSIGN( 8, 7) + ASSIGN( 8, 8)
     + ASSIGN( 8, 9) =     1
10]   ASSIGN( 9, 1) + ASSIGN( 9, 2) + ASSIGN( 9, 3) + ASSIGN( 9, 4)
     + ASSIGN( 9, 5) + ASSIGN( 9, 6) + ASSIGN( 9, 7) + ASSIGN( 9, 8)
     + ASSIGN( 9, 9) =     1
11]   ASSIGN( 10, 1) + ASSIGN( 10, 2) + ASSIGN( 10, 3) + ASSIGN( 10, 4)
     + ASSIGN( 10, 5) + ASSIGN( 10, 6) + ASSIGN( 10, 7)
     + ASSIGN( 10, 8) + ASSIGN( 10, 9) =     1
12]   ASSIGN( 11, 1) + ASSIGN( 11, 2) + ASSIGN( 11, 3) + ASSIGN( 11, 4)
     + ASSIGN( 11, 5) + ASSIGN( 11, 6) + ASSIGN( 11, 7)
     + ASSIGN( 11, 8) + ASSIGN( 11, 9) =     1
13]   ASSIGN( 12, 1) + ASSIGN( 12, 2) + ASSIGN( 12, 3) + ASSIGN( 12, 4)
     + ASSIGN( 12, 5) + ASSIGN( 12, 6) + ASSIGN( 12, 7)
     + ASSIGN( 12, 8) + ASSIGN( 12, 9) =     1
14]   ASSIGN( 13, 1) + ASSIGN( 13, 2) + ASSIGN( 13, 3) + ASSIGN( 13, 4)
     + ASSIGN( 13, 5) + ASSIGN( 13, 6) + ASSIGN( 13, 7)
     + ASSIGN( 13, 8) + ASSIGN( 13, 9) =     1
```

```
15]  ASSIGN( 14, 1) + ASSIGN( 14, 2) + ASSIGN( 14, 3) + ASSIGN( 14, 4)
      + ASSIGN( 14, 5) + ASSIGN( 14, 6) + ASSIGN( 14, 7)
      + ASSIGN( 14, 8) + ASSIGN( 14, 9) =    1
 16]  ASSIGN( 15, 1) + ASSIGN( 15, 2) + ASSIGN( 15, 3) + ASSIGN( 15, 4)
      + ASSIGN( 15, 5) + ASSIGN( 15, 6) + ASSIGN( 15, 7)
      + ASSIGN( 15, 8) + ASSIGN( 15, 9) =    1
 17]  ASSIGN( 16, 1) + ASSIGN( 16, 2) + ASSIGN( 16, 3) + ASSIGN( 16, 4)
      + ASSIGN( 16, 5) + ASSIGN( 16, 6) + ASSIGN( 16, 7)
      + ASSIGN( 16, 8) + ASSIGN( 16, 9) =    1
 18]  ASSIGN( 17, 1) + ASSIGN( 17, 2) + ASSIGN( 17, 3) + ASSIGN( 17, 4)
      + ASSIGN( 17, 5) + ASSIGN( 17, 6) + ASSIGN( 17, 7)
      + ASSIGN( 17, 8) + ASSIGN( 17, 9) =    1
 19]  ASSIGN( 18, 1) + ASSIGN( 18, 2) + ASSIGN( 18, 3) + ASSIGN( 18, 4)
      + ASSIGN( 18, 5) + ASSIGN( 18, 6) + ASSIGN( 18, 7)
      + ASSIGN( 18, 8) + ASSIGN( 18, 9) =    1
 20]  ASSIGN( 19, 1) + ASSIGN( 19, 2) + ASSIGN( 19, 3) + ASSIGN( 19, 4)
      + ASSIGN( 19, 5) + ASSIGN( 19, 6) + ASSIGN( 19, 7)
      + ASSIGN( 19, 8) + ASSIGN( 19, 9) =    1
 21]  ASSIGN( 20, 1) + ASSIGN( 20, 2) + ASSIGN( 20, 3) + ASSIGN( 20, 4)
      + ASSIGN( 20, 5) + ASSIGN( 20, 6) + ASSIGN( 20, 7)
      + ASSIGN( 20, 8) + ASSIGN( 20, 9) =    1
 22]  ASSIGN( 21, 1) + ASSIGN( 21, 2) + ASSIGN( 21, 3) + ASSIGN( 21, 4)
      + ASSIGN( 21, 5) + ASSIGN( 21, 6) + ASSIGN( 21, 7)
      + ASSIGN( 21, 8) + ASSIGN( 21, 9) =    1
 23]  ASSIGN( 22, 1) + ASSIGN( 22, 2) + ASSIGN( 22, 3) + ASSIGN( 22, 4)
      + ASSIGN( 22, 5) + ASSIGN( 22, 6) + ASSIGN( 22, 7)
      + ASSIGN( 22, 8) + ASSIGN( 22, 9) =    1
 24]  ASSIGN( 23, 1) + ASSIGN( 23, 2) + ASSIGN( 23, 3) + ASSIGN( 23, 4)
      + ASSIGN( 23, 5) + ASSIGN( 23, 6) + ASSIGN( 23, 7)
      + ASSIGN( 23, 8) + ASSIGN( 23, 9) =    1
 25]  ASSIGN( 24, 1) + ASSIGN( 24, 2) + ASSIGN( 24, 3) + ASSIGN( 24, 4)
      + ASSIGN( 24, 5) + ASSIGN( 24, 6) + ASSIGN( 24, 7)
      + ASSIGN( 24, 8) + ASSIGN( 24, 9) =    1
 26]  ASSIGN( 25, 1) + ASSIGN( 25, 2) + ASSIGN( 25, 3) + ASSIGN( 25, 4)
      + ASSIGN( 25, 5) + ASSIGN( 25, 6) + ASSIGN( 25, 7)
      + ASSIGN( 25, 8) + ASSIGN( 25, 9) =    1
 27]  ASSIGN( 26, 1) + ASSIGN( 26, 2) + ASSIGN( 26, 3) + ASSIGN( 26, 4)
      + ASSIGN( 26, 5) + ASSIGN( 26, 6) + ASSIGN( 26, 7)
      + ASSIGN( 26, 8) + ASSIGN( 26, 9) =    1
 28]  ASSIGN( 27, 1) + ASSIGN( 27, 2) + ASSIGN( 27, 3) + ASSIGN( 27, 4)
      + ASSIGN( 27, 5) + ASSIGN( 27, 6) + ASSIGN( 27, 7)
      + ASSIGN( 27, 8) + ASSIGN( 27, 9) =    1
 29]  ASSIGN( 28, 1) + ASSIGN( 28, 2) + ASSIGN( 28, 3) + ASSIGN( 28, 4)
      + ASSIGN( 28, 5) + ASSIGN( 28, 6) + ASSIGN( 28, 7)
      + ASSIGN( 28, 8) + ASSIGN( 28, 9) =    1
 30]  ASSIGN( 29, 1) + ASSIGN( 29, 2) + ASSIGN( 29, 3) + ASSIGN( 29, 4)
      + ASSIGN( 29, 5) + ASSIGN( 29, 6) + ASSIGN( 29, 7)
      + ASSIGN( 29, 8) + ASSIGN( 29, 9) =    1
 31]  ASSIGN( 30, 1) + ASSIGN( 30, 2) + ASSIGN( 30, 3) + ASSIGN( 30, 4)
      + ASSIGN( 30, 5) + ASSIGN( 30, 6) + ASSIGN( 30, 7)
      + ASSIGN( 30, 8) + ASSIGN( 30, 9) =    1
 32]  ASSIGN( 31, 1) + ASSIGN( 31, 2) + ASSIGN( 31, 3) + ASSIGN( 31, 4)
      + ASSIGN( 31, 5) + ASSIGN( 31, 6) + ASSIGN( 31, 7)
      + ASSIGN( 31, 8) + ASSIGN( 31, 9) =    1
 33]  ASSIGN( 32, 1) + ASSIGN( 32, 2) + ASSIGN( 32, 3) + ASSIGN( 32, 4)
      + ASSIGN( 32, 5) + ASSIGN( 32, 6) + ASSIGN( 32, 7)
      + ASSIGN( 32, 8) + ASSIGN( 32, 9) =    1
```

```
34]  ASSIGN( 33, 1) + ASSIGN( 33, 2) + ASSIGN( 33, 3) + ASSIGN( 33, 4)
     + ASSIGN( 33, 5) + ASSIGN( 33, 6) + ASSIGN( 33, 7)
     + ASSIGN( 33, 8) + ASSIGN( 33, 9) =     1
35]  ASSIGN( 34, 1) + ASSIGN( 34, 2) + ASSIGN( 34, 3) + ASSIGN( 34, 4)
     + ASSIGN( 34, 5) + ASSIGN( 34, 6) + ASSIGN( 34, 7)
     + ASSIGN( 34, 8) + ASSIGN( 34, 9) =     1
36]  ASSIGN( 35, 1) + ASSIGN( 35, 2) + ASSIGN( 35, 3) + ASSIGN( 35, 4)
     + ASSIGN( 35, 5) + ASSIGN( 35, 6) + ASSIGN( 35, 7)
     + ASSIGN( 35, 8) + ASSIGN( 35, 9) =     1
37]  ASSIGN( 36, 1) + ASSIGN( 36, 2) + ASSIGN( 36, 3) + ASSIGN( 36, 4)
     + ASSIGN( 36, 5) + ASSIGN( 36, 6) + ASSIGN( 36, 7)
     + ASSIGN( 36, 8) + ASSIGN( 36, 9) =     1
38]  ASSIGN( 37, 1) + ASSIGN( 37, 2) + ASSIGN( 37, 3) + ASSIGN( 37, 4)
     + ASSIGN( 37, 5) + ASSIGN( 37, 6) + ASSIGN( 37, 7)
     + ASSIGN( 37, 8) + ASSIGN( 37, 9) =     1
39]  ASSIGN( 38, 1) + ASSIGN( 38, 2) + ASSIGN( 38, 3) + ASSIGN( 38, 4)
     + ASSIGN( 38, 5) + ASSIGN( 38, 6) + ASSIGN( 38, 7)
     + ASSIGN( 38, 8) + ASSIGN( 38, 9) =     1
40]  ASSIGN( 39, 1) + ASSIGN( 39, 2) + ASSIGN( 39, 3) + ASSIGN( 39, 4)
     + ASSIGN( 39, 5) + ASSIGN( 39, 6) + ASSIGN( 39, 7)
     + ASSIGN( 39, 8) + ASSIGN( 39, 9) =     1
41]  ASSIGN( 40, 1) + ASSIGN( 40, 2) + ASSIGN( 40, 3) + ASSIGN( 40, 4)
     + ASSIGN( 40, 5) + ASSIGN( 40, 6) + ASSIGN( 40, 7)
     + ASSIGN( 40, 8) + ASSIGN( 40, 9) =     1
42]  ASSIGN( 41, 1) + ASSIGN( 41, 2) + ASSIGN( 41, 3) + ASSIGN( 41, 4)
     + ASSIGN( 41, 5) + ASSIGN( 41, 6) + ASSIGN( 41, 7)
     + ASSIGN( 41, 8) + ASSIGN( 41, 9) =     1
43]  ASSIGN( 42, 1) + ASSIGN( 42, 2) + ASSIGN( 42, 3) + ASSIGN( 42, 4)
     + ASSIGN( 42, 5) + ASSIGN( 42, 6) + ASSIGN( 42, 7)
     + ASSIGN( 42, 8) + ASSIGN( 42, 9) =     1
44]  ASSIGN( 43, 1) + ASSIGN( 43, 2) + ASSIGN( 43, 3) + ASSIGN( 43, 4)
     + ASSIGN( 43, 5) + ASSIGN( 43, 6) + ASSIGN( 43, 7)
     + ASSIGN( 43, 8) + ASSIGN( 43, 9) =     1
45]  ASSIGN( 44, 1) + ASSIGN( 44, 2) + ASSIGN( 44, 3) + ASSIGN( 44, 4)
     + ASSIGN( 44, 5) + ASSIGN( 44, 6) + ASSIGN( 44, 7)
     + ASSIGN( 44, 8) + ASSIGN( 44, 9) =     1
46]  ASSIGN( 45, 1) + ASSIGN( 45, 2) + ASSIGN( 45, 3) + ASSIGN( 45, 4)
     + ASSIGN( 45, 5) + ASSIGN( 45, 6) + ASSIGN( 45, 7)
     + ASSIGN( 45, 8) + ASSIGN( 45, 9) =     1
47]  ASSIGN( 46, 1) + ASSIGN( 46, 2) + ASSIGN( 46, 3) + ASSIGN( 46, 4)
     + ASSIGN( 46, 5) + ASSIGN( 46, 6) + ASSIGN( 46, 7)
     + ASSIGN( 46, 8) + ASSIGN( 46, 9) =     1
48]  ASSIGN( 47, 1) + ASSIGN( 47, 2) + ASSIGN( 47, 3) + ASSIGN( 47, 4)
     + ASSIGN( 47, 5) + ASSIGN( 47, 6) + ASSIGN( 47, 7)
     + ASSIGN( 47, 8) + ASSIGN( 47, 9) =     1
49]  ASSIGN( 48, 1) + ASSIGN( 48, 2) + ASSIGN( 48, 3) + ASSIGN( 48, 4)
     + ASSIGN( 48, 5) + ASSIGN( 48, 6) + ASSIGN( 48, 7)
     + ASSIGN( 48, 8) + ASSIGN( 48, 9) =     1
50]  ASSIGN( 49, 1) + ASSIGN( 49, 2) + ASSIGN( 49, 3) + ASSIGN( 49, 4)
     + ASSIGN( 49, 5) + ASSIGN( 49, 6) + ASSIGN( 49, 7)
     + ASSIGN( 49, 8) + ASSIGN( 49, 9) =     1
51]  ASSIGN( 50, 1) + ASSIGN( 50, 2) + ASSIGN( 50, 3) + ASSIGN( 50, 4)
     + ASSIGN( 50, 5) + ASSIGN( 50, 6) + ASSIGN( 50, 7)
     + ASSIGN( 50, 8) + ASSIGN( 50, 9) =     1
52]  ASSIGN( 51, 1) + ASSIGN( 51, 2) + ASSIGN( 51, 3) + ASSIGN( 51, 4)
     + ASSIGN( 51, 5) + ASSIGN( 51, 6) + ASSIGN( 51, 7)
     + ASSIGN( 51, 8) + ASSIGN( 51, 9) =     1
```

```
53]  ASSIGN( 52, 1) + ASSIGN( 52, 2) + ASSIGN( 52, 3) + ASSIGN( 52, 4)
     + ASSIGN( 52, 5) + ASSIGN( 52, 6) + ASSIGN( 52, 7)
     + ASSIGN( 52, 8) + ASSIGN( 52, 9) =    1
54]  ASSIGN( 53, 1) + ASSIGN( 53, 2) + ASSIGN( 53, 3) + ASSIGN( 53, 4)
     + ASSIGN( 53, 5) + ASSIGN( 53, 6) + ASSIGN( 53, 7)
     + ASSIGN( 53, 8) + ASSIGN( 53, 9) =    1
55]  ASSIGN( 54, 1) + ASSIGN( 54, 2) + ASSIGN( 54, 3) + ASSIGN( 54, 4)
     + ASSIGN( 54, 5) + ASSIGN( 54, 6) + ASSIGN( 54, 7)
     + ASSIGN( 54, 8) + ASSIGN( 54, 9) =    1
56]  ASSIGN( 55, 1) + ASSIGN( 55, 2) + ASSIGN( 55, 3) + ASSIGN( 55, 4)
     + ASSIGN( 55, 5) + ASSIGN( 55, 6) + ASSIGN( 55, 7)
     + ASSIGN( 55, 8) + ASSIGN( 55, 9) =    1
57]  ASSIGN( 56, 1) + ASSIGN( 56, 2) + ASSIGN( 56, 3) + ASSIGN( 56, 4)
     + ASSIGN( 56, 5) + ASSIGN( 56, 6) + ASSIGN( 56, 7)
     + ASSIGN( 56, 8) + ASSIGN( 56, 9) =    1
58]  ASSIGN( 57, 1) + ASSIGN( 57, 2) + ASSIGN( 57, 3) + ASSIGN( 57, 4)
     + ASSIGN( 57, 5) + ASSIGN( 57, 6) + ASSIGN( 57, 7)
     + ASSIGN( 57, 8) + ASSIGN( 57, 9) =    1
59]  ASSIGN( 1, 1) + ASSIGN( 2, 1) + ASSIGN( 3, 1) + ASSIGN( 4, 1)
     + ASSIGN( 5, 1) + ASSIGN( 6, 1) + ASSIGN( 7, 1) + ASSIGN( 8, 1)
     + ASSIGN( 9, 1) + ASSIGN( 10, 1) + ASSIGN( 11, 1)
     + ASSIGN( 12, 1) + ASSIGN( 13, 1) + ASSIGN( 14, 1)
     + ASSIGN( 15, 1) + ASSIGN( 16, 1) + ASSIGN( 17, 1)
     + ASSIGN( 18, 1) + ASSIGN( 19, 1) + ASSIGN( 20, 1)
     + ASSIGN( 21, 1) + ASSIGN( 22, 1) + ASSIGN( 23, 1)
     + ASSIGN( 24, 1) + ASSIGN( 25, 1) + ASSIGN( 26, 1)
     + ASSIGN( 27, 1) + ASSIGN( 28, 1) + ASSIGN( 29, 1)
     + ASSIGN( 30, 1) + ASSIGN( 31, 1) + ASSIGN( 32, 1)
     + ASSIGN( 33, 1) + ASSIGN( 34, 1) + ASSIGN( 35, 1)
     + ASSIGN( 36, 1) + ASSIGN( 37, 1) + ASSIGN( 38, 1)
     + ASSIGN( 39, 1) + ASSIGN( 40, 1) + ASSIGN( 41, 1)
     + ASSIGN( 42, 1) + ASSIGN( 43, 1) + ASSIGN( 44, 1)
     + ASSIGN( 45, 1) + ASSIGN( 46, 1) + ASSIGN( 47, 1)
     + ASSIGN( 48, 1) + ASSIGN( 49, 1) + ASSIGN( 50, 1)
     + ASSIGN( 51, 1) + ASSIGN( 52, 1) + ASSIGN( 53, 1)
     + ASSIGN( 54, 1) + ASSIGN( 55, 1) + ASSIGN( 56, 1)
     + ASSIGN( 57, 1) - 57 AIRCRAFTNUMBER( 1) <=    0
60]  ASSIGN( 1, 2) + ASSIGN( 2, 2) + ASSIGN( 3, 2) + ASSIGN( 4, 2)
     + ASSIGN( 5, 2) + ASSIGN( 6, 2) + ASSIGN( 7, 2) + ASSIGN( 8, 2)
     + ASSIGN( 9, 2) + ASSIGN( 10, 2) + ASSIGN( 11, 2)
     + ASSIGN( 12, 2) + ASSIGN( 13, 2) + ASSIGN( 14, 2)
     + ASSIGN( 15, 2) + ASSIGN( 16, 2) + ASSIGN( 17, 2)
     + ASSIGN( 18, 2) + ASSIGN( 19, 2) + ASSIGN( 20, 2)
     + ASSIGN( 21, 2) + ASSIGN( 22, 2) + ASSIGN( 23, 2)
     + ASSIGN( 24, 2) + ASSIGN( 25, 2) + ASSIGN( 26, 2)
     + ASSIGN( 27, 2) + ASSIGN( 28, 2) + ASSIGN( 29, 2)
     + ASSIGN( 30, 2) + ASSIGN( 31, 2) + ASSIGN( 32, 2)
     + ASSIGN( 33, 2) + ASSIGN( 34, 2) + ASSIGN( 35, 2)
     + ASSIGN( 36, 2) + ASSIGN( 37, 2) + ASSIGN( 38, 2)
     + ASSIGN( 39, 2) + ASSIGN( 40, 2) + ASSIGN( 41, 2)
     + ASSIGN( 42, 2) + ASSIGN( 43, 2) + ASSIGN( 44, 2)
     + ASSIGN( 45, 2) + ASSIGN( 46, 2) + ASSIGN( 47, 2)
     + ASSIGN( 48, 2) + ASSIGN( 49, 2) + ASSIGN( 50, 2)
     + ASSIGN( 51, 2) + ASSIGN( 52, 2) + ASSIGN( 53, 2)
     + ASSIGN( 54, 2) + ASSIGN( 55, 2) + ASSIGN( 56, 2)
     + ASSIGN( 57, 2) - 57 AIRCRAFTNUMBER( 2) <=    0
61]  ASSIGN( 1, 3) + ASSIGN( 2, 3) + ASSIGN( 3, 3) + ASSIGN( 4, 3)
```

```
     + ASSIGN( 5, 3) + ASSIGN( 6, 3) + ASSIGN( 7, 3) + ASSIGN( 8, 3)
     + ASSIGN( 9, 3) + ASSIGN( 10, 3) + ASSIGN( 11, 3)
     + ASSIGN( 12, 3) + ASSIGN( 13, 3) + ASSIGN( 14, 3)
     + ASSIGN( 15, 3) + ASSIGN( 16, 3) + ASSIGN( 17, 3)
     + ASSIGN( 18, 3) + ASSIGN( 19, 3) + ASSIGN( 20, 3)
     + ASSIGN( 21, 3) + ASSIGN( 22, 3) + ASSIGN( 23, 3)
     + ASSIGN( 24, 3) + ASSIGN( 25, 3) + ASSIGN( 26, 3)
     + ASSIGN( 27, 3) + ASSIGN( 28, 3) + ASSIGN( 29, 3)
     + ASSIGN( 30, 3) + ASSIGN( 31, 3) + ASSIGN( 32, 3)
     + ASSIGN( 33, 3) + ASSIGN( 34, 3) + ASSIGN( 35, 3)
     + ASSIGN( 36, 3) + ASSIGN( 37, 3) + ASSIGN( 38, 3)
     + ASSIGN( 39, 3) + ASSIGN( 40, 3) + ASSIGN( 41, 3)
     + ASSIGN( 42, 3) + ASSIGN( 43, 3) + ASSIGN( 44, 3)
     + ASSIGN( 45, 3) + ASSIGN( 46, 3) + ASSIGN( 47, 3)
     + ASSIGN( 48, 3) + ASSIGN( 49, 3) + ASSIGN( 50, 3)
     + ASSIGN( 51, 3) + ASSIGN( 52, 3) + ASSIGN( 53, 3)
     + ASSIGN( 54, 3) + ASSIGN( 55, 3) + ASSIGN( 56, 3)
     + ASSIGN( 57, 3) - 57 AIRCRAFTNUMBER( 3) <=    0
62]  ASSIGN( 1, 4) + ASSIGN( 2, 4) + ASSIGN( 3, 4) + ASSIGN( 4, 4)
     + ASSIGN( 5, 4) + ASSIGN( 6, 4) + ASSIGN( 7, 4) + ASSIGN( 8, 4)
     + ASSIGN( 9, 4) + ASSIGN( 10, 4) + ASSIGN( 11, 4)
     + ASSIGN( 12, 4) + ASSIGN( 13, 4) + ASSIGN( 14, 4)
     + ASSIGN( 15, 4) + ASSIGN( 16, 4) + ASSIGN( 17, 4)
     + ASSIGN( 18, 4) + ASSIGN( 19, 4) + ASSIGN( 20, 4)
     + ASSIGN( 21, 4) + ASSIGN( 22, 4) + ASSIGN( 23, 4)
     + ASSIGN( 24, 4) + ASSIGN( 25, 4) + ASSIGN( 26, 4)
     + ASSIGN( 27, 4) + ASSIGN( 28, 4) + ASSIGN( 29, 4)
     + ASSIGN( 30, 4) + ASSIGN( 31, 4) + ASSIGN( 32, 4)
     + ASSIGN( 33, 4) + ASSIGN( 34, 4) + ASSIGN( 35, 4)
     + ASSIGN( 36, 4) + ASSIGN( 37, 4) + ASSIGN( 38, 4)
     + ASSIGN( 39, 4) + ASSIGN( 40, 4) + ASSIGN( 41, 4)
     + ASSIGN( 42, 4) + ASSIGN( 43, 4) + ASSIGN( 44, 4)
     + ASSIGN( 45, 4) + ASSIGN( 46, 4) + ASSIGN( 47, 4)
     + ASSIGN( 48, 4) + ASSIGN( 49, 4) + ASSIGN( 50, 4)
     + ASSIGN( 51, 4) + ASSIGN( 52, 4) + ASSIGN( 53, 4)
     + ASSIGN( 54, 4) + ASSIGN( 55, 4) + ASSIGN( 56, 4)
     + ASSIGN( 57, 4) - 57 AIRCRAFTNUMBER( 4) <=    0
63]  ASSIGN( 1, 5) + ASSIGN( 2, 5) + ASSIGN( 3, 5) + ASSIGN( 4, 5)
     + ASSIGN( 5, 5) + ASSIGN( 6, 5) + ASSIGN( 7, 5) + ASSIGN( 8, 5)
     + ASSIGN( 9, 5) + ASSIGN( 10, 5) + ASSIGN( 11, 5)
     + ASSIGN( 12, 5) + ASSIGN( 13, 5) + ASSIGN( 14, 5)
     + ASSIGN( 15, 5) + ASSIGN( 16, 5) + ASSIGN( 17, 5)
     + ASSIGN( 18, 5) + ASSIGN( 19, 5) + ASSIGN( 20, 5)
     + ASSIGN( 21, 5) + ASSIGN( 22, 5) + ASSIGN( 23, 5)
     + ASSIGN( 24, 5) + ASSIGN( 25, 5) + ASSIGN( 26, 5)
     + ASSIGN( 27, 5) + ASSIGN( 28, 5) + ASSIGN( 29, 5)
     + ASSIGN( 30, 5) + ASSIGN( 31, 5) + ASSIGN( 32, 5)
     + ASSIGN( 33, 5) + ASSIGN( 34, 5) + ASSIGN( 35, 5)
     + ASSIGN( 36, 5) + ASSIGN( 37, 5) + ASSIGN( 38, 5)
     + ASSIGN( 39, 5) + ASSIGN( 40, 5) + ASSIGN( 41, 5)
     + ASSIGN( 42, 5) + ASSIGN( 43, 5) + ASSIGN( 44, 5)
     + ASSIGN( 45, 5) + ASSIGN( 46, 5) + ASSIGN( 47, 5)
     + ASSIGN( 48, 5) + ASSIGN( 49, 5) + ASSIGN( 50, 5)
     + ASSIGN( 51, 5) + ASSIGN( 52, 5) + ASSIGN( 53, 5)
     + ASSIGN( 54, 5) + ASSIGN( 55, 5) + ASSIGN( 56, 5)
     + ASSIGN( 57, 5) - 57 AIRCRAFTNUMBER( 5) <=    0
```

```
64]  ASSIGN( 1, 6) + ASSIGN( 2, 6) + ASSIGN( 3, 6) + ASSIGN( 4, 6)
      + ASSIGN( 5, 6) + ASSIGN( 6, 6) + ASSIGN( 7, 6) + ASSIGN( 8, 6)
      + ASSIGN( 9, 6) + ASSIGN( 10, 6) + ASSIGN( 11, 6)
      + ASSIGN( 12, 6) + ASSIGN( 13, 6) + ASSIGN( 14, 6)
      + ASSIGN( 15, 6) + ASSIGN( 16, 6) + ASSIGN( 17, 6)
      + ASSIGN( 18, 6) + ASSIGN( 19, 6) + ASSIGN( 20, 6)
      + ASSIGN( 21, 6) + ASSIGN( 22, 6) + ASSIGN( 23, 6)
      + ASSIGN( 24, 6) + ASSIGN( 25, 6) + ASSIGN( 26, 6)
      + ASSIGN( 27, 6) + ASSIGN( 28, 6) + ASSIGN( 29, 6)
      + ASSIGN( 30, 6) + ASSIGN( 31, 6) + ASSIGN( 32, 6)
      + ASSIGN( 33, 6) + ASSIGN( 34, 6) + ASSIGN( 35, 6)
      + ASSIGN( 36, 6) + ASSIGN( 37, 6) + ASSIGN( 38, 6)
      + ASSIGN( 39, 6) + ASSIGN( 40, 6) + ASSIGN( 41, 6)
      + ASSIGN( 42, 6) + ASSIGN( 43, 6) + ASSIGN( 44, 6)
      + ASSIGN( 45, 6) + ASSIGN( 46, 6) + ASSIGN( 47, 6)
      + ASSIGN( 48, 6) + ASSIGN( 49, 6) + ASSIGN( 50, 6)
      + ASSIGN( 51, 6) + ASSIGN( 52, 6) + ASSIGN( 53, 6)
      + ASSIGN( 54, 6) + ASSIGN( 55, 6) + ASSIGN( 56, 6)
      + ASSIGN( 57, 6) - 57 AIRCRAFTNUMBER( 6) <=   0
 65]  ASSIGN( 1, 7) + ASSIGN( 2, 7) + ASSIGN( 3, 7) + ASSIGN( 4, 7)
      + ASSIGN( 5, 7) + ASSIGN( 6, 7) + ASSIGN( 7, 7) + ASSIGN( 8, 7)
      + ASSIGN( 9, 7) + ASSIGN( 10, 7) + ASSIGN( 11, 7)
      + ASSIGN( 12, 7) + ASSIGN( 13, 7) + ASSIGN( 14, 7)
      + ASSIGN( 15, 7) + ASSIGN( 16, 7) + ASSIGN( 17, 7)
      + ASSIGN( 18, 7) + ASSIGN( 19, 7) + ASSIGN( 20, 7)
      + ASSIGN( 21, 7) + ASSIGN( 22, 7) + ASSIGN( 23, 7)
      + ASSIGN( 24, 7) + ASSIGN( 25, 7) + ASSIGN( 26, 7)
      + ASSIGN( 27, 7) + ASSIGN( 28, 7) + ASSIGN( 29, 7)
      + ASSIGN( 30, 7) + ASSIGN( 31, 7) + ASSIGN( 32, 7)
      + ASSIGN( 33, 7) + ASSIGN( 34, 7) + ASSIGN( 35, 7)
      + ASSIGN( 36, 7) + ASSIGN( 37, 7) + ASSIGN( 38, 7)
      + ASSIGN( 39, 7) + ASSIGN( 40, 7) + ASSIGN( 41, 7)
      + ASSIGN( 42, 7) + ASSIGN( 43, 7) + ASSIGN( 44, 7)
      + ASSIGN( 45, 7) + ASSIGN( 46, 7) + ASSIGN( 47, 7)
      + ASSIGN( 48, 7) + ASSIGN( 49, 7) + ASSIGN( 50, 7)
      + ASSIGN( 51, 7) + ASSIGN( 52, 7) + ASSIGN( 53, 7)
      + ASSIGN( 54, 7) + ASSIGN( 55, 7) + ASSIGN( 56, 7)
      + ASSIGN( 57, 7) - 57 AIRCRAFTNUMBER( 7) <=   0
 66]  ASSIGN( 1, 8) + ASSIGN( 2, 8) + ASSIGN( 3, 8) + ASSIGN( 4, 8)
      + ASSIGN( 5, 8) + ASSIGN( 6, 8) + ASSIGN( 7, 8) + ASSIGN( 8, 8)
      + ASSIGN( 9, 8) + ASSIGN( 10, 8) + ASSIGN( 11, 8)
      + ASSIGN( 12, 8) + ASSIGN( 13, 8) + ASSIGN( 14, 8)
      + ASSIGN( 15, 8) + ASSIGN( 16, 8) + ASSIGN( 17, 8)
      + ASSIGN( 18, 8) + ASSIGN( 19, 8) + ASSIGN( 20, 8)
      + ASSIGN( 21, 8) + ASSIGN( 22, 8) + ASSIGN( 23, 8)
      + ASSIGN( 24, 8) + ASSIGN( 25, 8) + ASSIGN( 26, 8)
      + ASSIGN( 27, 8) + ASSIGN( 28, 8) + ASSIGN( 29, 8)
      + ASSIGN( 30, 8) + ASSIGN( 31, 8) + ASSIGN( 32, 8)
      + ASSIGN( 33, 8) + ASSIGN( 34, 8) + ASSIGN( 35, 8)
      + ASSIGN( 36, 8) + ASSIGN( 37, 8) + ASSIGN( 38, 8)
      + ASSIGN( 39, 8) + ASSIGN( 40, 8) + ASSIGN( 41, 8)
      + ASSIGN( 42, 8) + ASSIGN( 43, 8) + ASSIGN( 44, 8)
      + ASSIGN( 45, 8) + ASSIGN( 46, 8) + ASSIGN( 47, 8)
      + ASSIGN( 48, 8) + ASSIGN( 49, 8) + ASSIGN( 50, 8)
      + ASSIGN( 51, 8) + ASSIGN( 52, 8) + ASSIGN( 53, 8)
      + ASSIGN( 54, 8) + ASSIGN( 55, 8) + ASSIGN( 56, 8)
      + ASSIGN( 57, 8) - 57 AIRCRAFTNUMBER( 8) <=   0
```

```
67]  ASSIGN( 1, 9) + ASSIGN( 2, 9) + ASSIGN( 3, 9) + ASSIGN( 4, 9)
     + ASSIGN( 5, 9) + ASSIGN( 6, 9) + ASSIGN( 7, 9) + ASSIGN( 8, 9)
     + ASSIGN( 9, 9) + ASSIGN( 10, 9) + ASSIGN( 11, 9)
     + ASSIGN( 12, 9) + ASSIGN( 13, 9) + ASSIGN( 14, 9)
     + ASSIGN( 15, 9) + ASSIGN( 16, 9) + ASSIGN( 17, 9)
     + ASSIGN( 18, 9) + ASSIGN( 19, 9) + ASSIGN( 20, 9)
     + ASSIGN( 21, 9) + ASSIGN( 22, 9) + ASSIGN( 23, 9)
     + ASSIGN( 24, 9) + ASSIGN( 25, 9) + ASSIGN( 26, 9)
     + ASSIGN( 27, 9) + ASSIGN( 28, 9) + ASSIGN( 29, 9)
     + ASSIGN( 30, 9) + ASSIGN( 31, 9) + ASSIGN( 32, 9)
     + ASSIGN( 33, 9) + ASSIGN( 34, 9) + ASSIGN( 35, 9)
     + ASSIGN( 36, 9) + ASSIGN( 37, 9) + ASSIGN( 38, 9)
     + ASSIGN( 39, 9) + ASSIGN( 40, 9) + ASSIGN( 41, 9)
     + ASSIGN( 42, 9) + ASSIGN( 43, 9) + ASSIGN( 44, 9)
     + ASSIGN( 45, 9) + ASSIGN( 46, 9) + ASSIGN( 47, 9)
     + ASSIGN( 48, 9) + ASSIGN( 49, 9) + ASSIGN( 50, 9)
     + ASSIGN( 51, 9) + ASSIGN( 52, 9) + ASSIGN( 53, 9)
     + ASSIGN( 54, 9) + ASSIGN( 55, 9) + ASSIGN( 56, 9)
     + ASSIGN( 57, 9) - 57 AIRCRAFTNUMBER( 9) <=   0
68]  495 ASSIGN( 1, 1) + 810 ASSIGN( 2, 1) + 735 ASSIGN( 3, 1)
     + 435 ASSIGN( 4, 1) + 855 ASSIGN( 5, 1) + 555 ASSIGN( 6, 1)
     + 495 ASSIGN( 7, 1) + 140 ASSIGN( 8, 1) + 140 ASSIGN( 9, 1)
     + 140 ASSIGN( 10, 1) + 140 ASSIGN( 11, 1) + 210 ASSIGN( 12, 1)
     + 210 ASSIGN( 13, 1) + 210 ASSIGN( 14, 1) + 210 ASSIGN( 15, 1)
     + 210 ASSIGN( 16, 1) + 210 ASSIGN( 17, 1) + 210 ASSIGN( 18, 1)
     + 195 ASSIGN( 19, 1) + 195 ASSIGN( 20, 1) + 195 ASSIGN( 21, 1)
     + 195 ASSIGN( 22, 1) + 195 ASSIGN( 23, 1) + 195 ASSIGN( 24, 1)
     + 195 ASSIGN( 25, 1) + 105 ASSIGN( 26, 1) + 105 ASSIGN( 27, 1)
     + 160 ASSIGN( 28, 1) + 160 ASSIGN( 29, 1) + 160 ASSIGN( 30, 1)
     + 160 ASSIGN( 31, 1) + 255 ASSIGN( 32, 1) + 255 ASSIGN( 33, 1)
     + 255 ASSIGN( 34, 1) + 255 ASSIGN( 35, 1) + 255 ASSIGN( 36, 1)
     + 255 ASSIGN( 37, 1) + 225 ASSIGN( 38, 1) + 225 ASSIGN( 39, 1)
     + 225 ASSIGN( 40, 1) + 225 ASSIGN( 41, 1) + 225 ASSIGN( 42, 1)
     + 160 ASSIGN( 43, 1) + 160 ASSIGN( 44, 1) + 160 ASSIGN( 45, 1)
     + 160 ASSIGN( 46, 1) + 160 ASSIGN( 47, 1) + 160 ASSIGN( 48, 1)
     + 160 ASSIGN( 49, 1) + 330 ASSIGN( 50, 1) + 258 ASSIGN( 51, 1)
     + 294 ASSIGN( 52, 1) + 288 ASSIGN( 53, 1) + 342 ASSIGN( 54, 1)
     + 228 ASSIGN( 55, 1) + 204 ASSIGN( 56, 1) + 306 ASSIGN( 57, 1)
     <=   1800
69]  495 ASSIGN( 1, 2) + 810 ASSIGN( 2, 2) + 735 ASSIGN( 3, 2)
     + 435 ASSIGN( 4, 2) + 855 ASSIGN( 5, 2) + 555 ASSIGN( 6, 2)
     + 495 ASSIGN( 7, 2) + 140 ASSIGN( 8, 2) + 140 ASSIGN( 9, 2)
     + 140 ASSIGN( 10, 2) + 140 ASSIGN( 11, 2) + 210 ASSIGN( 12, 2)
     + 210 ASSIGN( 13, 2) + 210 ASSIGN( 14, 2) + 210 ASSIGN( 15, 2)
     + 210 ASSIGN( 16, 2) + 210 ASSIGN( 17, 2) + 210 ASSIGN( 18, 2)
     + 195 ASSIGN( 19, 2) + 195 ASSIGN( 20, 2) + 195 ASSIGN( 21, 2)
     + 195 ASSIGN( 22, 2) + 195 ASSIGN( 23, 2) + 195 ASSIGN( 24, 2)
     + 195 ASSIGN( 25, 2) + 105 ASSIGN( 26, 2) + 105 ASSIGN( 27, 2)
     + 160 ASSIGN( 28, 2) + 160 ASSIGN( 29, 2) + 160 ASSIGN( 30, 2)
     + 160 ASSIGN( 31, 2) + 255 ASSIGN( 32, 2) + 255 ASSIGN( 33, 2)
     + 255 ASSIGN( 34, 2) + 255 ASSIGN( 35, 2) + 255 ASSIGN( 36, 2)
     + 255 ASSIGN( 37, 2) + 225 ASSIGN( 38, 2) + 225 ASSIGN( 39, 2)
     + 225 ASSIGN( 40, 2) + 225 ASSIGN( 41, 2) + 225 ASSIGN( 42, 2)
     + 160 ASSIGN( 43, 2) + 160 ASSIGN( 44, 2) + 160 ASSIGN( 45, 2)
     + 160 ASSIGN( 46, 2) + 160 ASSIGN( 47, 2) + 160 ASSIGN( 48, 2)
     + 160 ASSIGN( 49, 2) + 330 ASSIGN( 50, 2) + 258 ASSIGN( 51, 2)
     + 294 ASSIGN( 52, 2) + 288 ASSIGN( 53, 2) + 342 ASSIGN( 54, 2)
```

```
    + 228 ASSIGN( 55, 2) + 204 ASSIGN( 56, 2) + 306 ASSIGN( 57, 2)
    <=   1800
70] 495 ASSIGN( 1, 3) + 810 ASSIGN( 2, 3) + 735 ASSIGN( 3, 3)
    + 435 ASSIGN( 4, 3) + 855 ASSIGN( 5, 3) + 555 ASSIGN( 6, 3)
    + 495 ASSIGN( 7, 3) + 140 ASSIGN( 8, 3) + 140 ASSIGN( 9, 3)
    + 140 ASSIGN( 10, 3) + 140 ASSIGN( 11, 3) + 210 ASSIGN( 12, 3)
    + 210 ASSIGN( 13, 3) + 210 ASSIGN( 14, 3) + 210 ASSIGN( 15, 3)
    + 210 ASSIGN( 16, 3) + 210 ASSIGN( 17, 3) + 210 ASSIGN( 18, 3)
    + 195 ASSIGN( 19, 3) + 195 ASSIGN( 20, 3) + 195 ASSIGN( 21, 3)
    + 195 ASSIGN( 22, 3) + 195 ASSIGN( 23, 3) + 195 ASSIGN( 24, 3)
    + 195 ASSIGN( 25, 3) + 105 ASSIGN( 26, 3) + 105 ASSIGN( 27, 3)
    + 160 ASSIGN( 28, 3) + 160 ASSIGN( 29, 3) + 160 ASSIGN( 30, 3)
    + 160 ASSIGN( 31, 3) + 255 ASSIGN( 32, 3) + 255 ASSIGN( 33, 3)
    + 255 ASSIGN( 34, 3) + 255 ASSIGN( 35, 3) + 255 ASSIGN( 36, 3)
    + 255 ASSIGN( 37, 3) + 225 ASSIGN( 38, 3) + 225 ASSIGN( 39, 3)
    + 225 ASSIGN( 40, 3) + 225 ASSIGN( 41, 3) + 225 ASSIGN( 42, 3)
    + 160 ASSIGN( 43, 3) + 160 ASSIGN( 44, 3) + 160 ASSIGN( 45, 3)
    + 160 ASSIGN( 46, 3) + 160 ASSIGN( 47, 3) + 160 ASSIGN( 48, 3)
    + 160 ASSIGN( 49, 3) + 330 ASSIGN( 50, 3) + 258 ASSIGN( 51, 3)
    + 294 ASSIGN( 52, 3) + 288 ASSIGN( 53, 3) + 342 ASSIGN( 54, 3)
    + 228 ASSIGN( 55, 3) + 204 ASSIGN( 56, 3) + 306 ASSIGN( 57, 3)
    <=   1800
71] 495 ASSIGN( 1, 4) + 810 ASSIGN( 2, 4) + 735 ASSIGN( 3, 4)
    + 435 ASSIGN( 4, 4) + 855 ASSIGN( 5, 4) + 555 ASSIGN( 6, 4)
    + 495 ASSIGN( 7, 4) + 140 ASSIGN( 8, 4) + 140 ASSIGN( 9, 4)
    + 140 ASSIGN( 10, 4) + 140 ASSIGN( 11, 4) + 210 ASSIGN( 12, 4)
    + 210 ASSIGN( 13, 4) + 210 ASSIGN( 14, 4) + 210 ASSIGN( 15, 4)
    + 210 ASSIGN( 16, 4) + 210 ASSIGN( 17, 4) + 210 ASSIGN( 18, 4)
    + 195 ASSIGN( 19, 4) + 195 ASSIGN( 20, 4) + 195 ASSIGN( 21, 4)
    + 195 ASSIGN( 22, 4) + 195 ASSIGN( 23, 4) + 195 ASSIGN( 24, 4)
    + 195 ASSIGN( 25, 4) + 105 ASSIGN( 26, 4) + 105 ASSIGN( 27, 4)
    + 160 ASSIGN( 28, 4) + 160 ASSIGN( 29, 4) + 160 ASSIGN( 30, 4)
    + 160 ASSIGN( 31, 4) + 255 ASSIGN( 32, 4) + 255 ASSIGN( 33, 4)
    + 255 ASSIGN( 34, 4) + 255 ASSIGN( 35, 4) + 255 ASSIGN( 36, 4)
    + 255 ASSIGN( 37, 4) + 225 ASSIGN( 38, 4) + 225 ASSIGN( 39, 4)
    + 225 ASSIGN( 40, 4) + 225 ASSIGN( 41, 4) + 225 ASSIGN( 42, 4)
    + 160 ASSIGN( 43, 4) + 160 ASSIGN( 44, 4) + 160 ASSIGN( 45, 4)
    + 160 ASSIGN( 46, 4) + 160 ASSIGN( 47, 4) + 160 ASSIGN( 48, 4)
    + 160 ASSIGN( 49, 4) + 330 ASSIGN( 50, 4) + 258 ASSIGN( 51, 4)
    + 294 ASSIGN( 52, 4) + 288 ASSIGN( 53, 4) + 342 ASSIGN( 54, 4)
    + 228 ASSIGN( 55, 4) + 204 ASSIGN( 56, 4) + 306 ASSIGN( 57, 4)
    <=   1800
72] 495 ASSIGN( 1, 5) + 810 ASSIGN( 2, 5) + 735 ASSIGN( 3, 5)
    + 435 ASSIGN( 4, 5) + 855 ASSIGN( 5, 5) + 555 ASSIGN( 6, 5)
    + 495 ASSIGN( 7, 5) + 140 ASSIGN( 8, 5) + 140 ASSIGN( 9, 5)
    + 140 ASSIGN( 10, 5) + 140 ASSIGN( 11, 5) + 210 ASSIGN( 12, 5)
    + 210 ASSIGN( 13, 5) + 210 ASSIGN( 14, 5) + 210 ASSIGN( 15, 5)
    + 210 ASSIGN( 16, 5) + 210 ASSIGN( 17, 5) + 210 ASSIGN( 18, 5)
    + 195 ASSIGN( 19, 5) + 195 ASSIGN( 20, 5) + 195 ASSIGN( 21, 5)
    + 195 ASSIGN( 22, 5) + 195 ASSIGN( 23, 5) + 195 ASSIGN( 24, 5)
    + 195 ASSIGN( 25, 5) + 105 ASSIGN( 26, 5) + 105 ASSIGN( 27, 5)
    + 160 ASSIGN( 28, 5) + 160 ASSIGN( 29, 5) + 160 ASSIGN( 30, 5)
    + 160 ASSIGN( 31, 5) + 255 ASSIGN( 32, 5) + 255 ASSIGN( 33, 5)
    + 255 ASSIGN( 34, 5) + 255 ASSIGN( 35, 5) + 255 ASSIGN( 36, 5)
    + 255 ASSIGN( 37, 5) + 225 ASSIGN( 38, 5) + 225 ASSIGN( 39, 5)
    + 225 ASSIGN( 40, 5) + 225 ASSIGN( 41, 5) + 225 ASSIGN( 42, 5)
    + 160 ASSIGN( 43, 5) + 160 ASSIGN( 44, 5) + 160 ASSIGN( 45, 5)
```

```
     + 160 ASSIGN( 46, 5) + 160 ASSIGN( 47, 5) + 160 ASSIGN( 48, 5)
     + 160 ASSIGN( 49, 5) + 330 ASSIGN( 50, 5) + 258 ASSIGN( 51, 5)
     + 294 ASSIGN( 52, 5) + 288 ASSIGN( 53, 5) + 342 ASSIGN( 54, 5)
     + 228 ASSIGN( 55, 5) + 204 ASSIGN( 56, 5) + 306 ASSIGN( 57, 5)
     <=    1800
73]  495 ASSIGN( 1, 6) + 810 ASSIGN( 2, 6) + 735 ASSIGN( 3, 6)
     + 435 ASSIGN( 4, 6) + 855 ASSIGN( 5, 6) + 555 ASSIGN( 6, 6)
     + 495 ASSIGN( 7, 6) + 140 ASSIGN( 8, 6) + 140 ASSIGN( 9, 6)
     + 140 ASSIGN( 10, 6) + 140 ASSIGN( 11, 6) + 210 ASSIGN( 12, 6)
     + 210 ASSIGN( 13, 6) + 210 ASSIGN( 14, 6) + 210 ASSIGN( 15, 6)
     + 210 ASSIGN( 16, 6) + 210 ASSIGN( 17, 6) + 210 ASSIGN( 18, 6)
     + 195 ASSIGN( 19, 6) + 195 ASSIGN( 20, 6) + 195 ASSIGN( 21, 6)
     + 195 ASSIGN( 22, 6) + 195 ASSIGN( 23, 6) + 195 ASSIGN( 24, 6)
     + 195 ASSIGN( 25, 6) + 105 ASSIGN( 26, 6) + 105 ASSIGN( 27, 6)
     + 160 ASSIGN( 28, 6) + 160 ASSIGN( 29, 6) + 160 ASSIGN( 30, 6)
     + 160 ASSIGN( 31, 6) + 255 ASSIGN( 32, 6) + 255 ASSIGN( 33, 6)
     + 255 ASSIGN( 34, 6) + 255 ASSIGN( 35, 6) + 255 ASSIGN( 36, 6)
     + 255 ASSIGN( 37, 6) + 225 ASSIGN( 38, 6) + 225 ASSIGN( 39, 6)
     + 225 ASSIGN( 40, 6) + 225 ASSIGN( 41, 6) + 225 ASSIGN( 42, 6)
     + 160 ASSIGN( 43, 6) + 160 ASSIGN( 44, 6) + 160 ASSIGN( 45, 6)
     + 160 ASSIGN( 46, 6) + 160 ASSIGN( 47, 6) + 160 ASSIGN( 48, 6)
     + 160 ASSIGN( 49, 6) + 330 ASSIGN( 50, 6) + 258 ASSIGN( 51, 6)
     + 294 ASSIGN( 52, 6) + 288 ASSIGN( 53, 6) + 342 ASSIGN( 54, 6)
     + 228 ASSIGN( 55, 6) + 204 ASSIGN( 56, 6) + 306 ASSIGN( 57, 6)
     <=    1800
74]  495 ASSIGN( 1, 7) + 810 ASSIGN( 2, 7) + 735 ASSIGN( 3, 7)
     + 435 ASSIGN( 4, 7) + 855 ASSIGN( 5, 7) + 555 ASSIGN( 6, 7)
     + 495 ASSIGN( 7, 7) + 140 ASSIGN( 8, 7) + 140 ASSIGN( 9, 7)
     + 140 ASSIGN( 10, 7) + 140 ASSIGN( 11, 7) + 210 ASSIGN( 12, 7)
     + 210 ASSIGN( 13, 7) + 210 ASSIGN( 14, 7) + 210 ASSIGN( 15, 7)
     + 210 ASSIGN( 16, 7) + 210 ASSIGN( 17, 7) + 210 ASSIGN( 18, 7)
     + 195 ASSIGN( 19, 7) + 195 ASSIGN( 20, 7) + 195 ASSIGN( 21, 7)
     + 195 ASSIGN( 22, 7) + 195 ASSIGN( 23, 7) + 195 ASSIGN( 24, 7)
     + 195 ASSIGN( 25, 7) + 105 ASSIGN( 26, 7) + 105 ASSIGN( 27, 7)
     + 160 ASSIGN( 28, 7) + 160 ASSIGN( 29, 7) + 160 ASSIGN( 30, 7)
     + 160 ASSIGN( 31, 7) + 255 ASSIGN( 32, 7) + 255 ASSIGN( 33, 7)
     + 255 ASSIGN( 34, 7) + 255 ASSIGN( 35, 7) + 255 ASSIGN( 36, 7)
     + 255 ASSIGN( 37, 7) + 225 ASSIGN( 38, 7) + 225 ASSIGN( 39, 7)
     + 225 ASSIGN( 40, 7) + 225 ASSIGN( 41, 7) + 225 ASSIGN( 42, 7)
     + 160 ASSIGN( 43, 7) + 160 ASSIGN( 44, 7) + 160 ASSIGN( 45, 7)
     + 160 ASSIGN( 46, 7) + 160 ASSIGN( 47, 7) + 160 ASSIGN( 48, 7)
     + 160 ASSIGN( 49, 7) + 330 ASSIGN( 50, 7) + 258 ASSIGN( 51, 7)
     + 294 ASSIGN( 52, 7) + 288 ASSIGN( 53, 7) + 342 ASSIGN( 54, 7)
     + 228 ASSIGN( 55, 7) + 204 ASSIGN( 56, 7) + 306 ASSIGN( 57, 7)
     <=    1800
75]  495 ASSIGN( 1, 8) + 810 ASSIGN( 2, 8) + 735 ASSIGN( 3, 8)
     + 435 ASSIGN( 4, 8) + 855 ASSIGN( 5, 8) + 555 ASSIGN( 6, 8)
     + 495 ASSIGN( 7, 8) + 140 ASSIGN( 8, 8) + 140 ASSIGN( 9, 8)
     + 140 ASSIGN( 10, 8) + 140 ASSIGN( 11, 8) + 210 ASSIGN( 12, 8)
     + 210 ASSIGN( 13, 8) + 210 ASSIGN( 14, 8) + 210 ASSIGN( 15, 8)
     + 210 ASSIGN( 16, 8) + 210 ASSIGN( 17, 8) + 210 ASSIGN( 18, 8)
     + 195 ASSIGN( 19, 8) + 195 ASSIGN( 20, 8) + 195 ASSIGN( 21, 8)
     + 195 ASSIGN( 22, 8) + 195 ASSIGN( 23, 8) + 195 ASSIGN( 24, 8)
     + 195 ASSIGN( 25, 8) + 105 ASSIGN( 26, 8) + 105 ASSIGN( 27, 8)
     + 160 ASSIGN( 28, 8) + 160 ASSIGN( 29, 8) + 160 ASSIGN( 30, 8)
     + 160 ASSIGN( 31, 8) + 255 ASSIGN( 32, 8) + 255 ASSIGN( 33, 8)
     + 255 ASSIGN( 34, 8) + 255 ASSIGN( 35, 8) + 255 ASSIGN( 36, 8)
```

```
     + 255 ASSIGN( 37, 8) + 225 ASSIGN( 38, 8) + 225 ASSIGN( 39, 8)
     + 225 ASSIGN( 40, 8) + 225 ASSIGN( 41, 8) + 225 ASSIGN( 42, 8)
     + 160 ASSIGN( 43, 8) + 160 ASSIGN( 44, 8) + 160 ASSIGN( 45, 8)
     + 160 ASSIGN( 46, 8) + 160 ASSIGN( 47, 8) + 160 ASSIGN( 48, 8)
     + 160 ASSIGN( 49, 8) + 330 ASSIGN( 50, 8) + 258 ASSIGN( 51, 8)
     + 294 ASSIGN( 52, 8) + 288 ASSIGN( 53, 8) + 342 ASSIGN( 54, 8)
     + 228 ASSIGN( 55, 8) + 204 ASSIGN( 56, 8) + 306 ASSIGN( 57, 8)
     <=    1800
76]  495 ASSIGN( 1, 9) + 810 ASSIGN( 2, 9) + 735 ASSIGN( 3, 9)
     + 435 ASSIGN( 4, 9) + 855 ASSIGN( 5, 9) + 555 ASSIGN( 6, 9)
     + 495 ASSIGN( 7, 9) + 140 ASSIGN( 8, 9) + 140 ASSIGN( 9, 9)
     + 140 ASSIGN( 10, 9) + 140 ASSIGN( 11, 9) + 210 ASSIGN( 12, 9)
     + 210 ASSIGN( 13, 9) + 210 ASSIGN( 14, 9) + 210 ASSIGN( 15, 9)
     + 210 ASSIGN( 16, 9) + 210 ASSIGN( 17, 9) + 210 ASSIGN( 18, 9)
     + 195 ASSIGN( 19, 9) + 195 ASSIGN( 20, 9) + 195 ASSIGN( 21, 9)
     + 195 ASSIGN( 22, 9) + 195 ASSIGN( 23, 9) + 195 ASSIGN( 24, 9)
     + 195 ASSIGN( 25, 9) + 105 ASSIGN( 26, 9) + 105 ASSIGN( 27, 9)
     + 160 ASSIGN( 28, 9) + 160 ASSIGN( 29, 9) + 160 ASSIGN( 30, 9)
     + 160 ASSIGN( 31, 9) + 255 ASSIGN( 32, 9) + 255 ASSIGN( 33, 9)
     + 255 ASSIGN( 34, 9) + 255 ASSIGN( 35, 9) + 255 ASSIGN( 36, 9)
     + 255 ASSIGN( 37, 9) + 225 ASSIGN( 38, 9) + 225 ASSIGN( 39, 9)
     + 225 ASSIGN( 40, 9) + 225 ASSIGN( 41, 9) + 225 ASSIGN( 42, 9)
     + 160 ASSIGN( 43, 9) + 160 ASSIGN( 44, 9) + 160 ASSIGN( 45, 9)
     + 160 ASSIGN( 46, 9) + 160 ASSIGN( 47, 9) + 160 ASSIGN( 48, 9)
     + 160 ASSIGN( 49, 9) + 330 ASSIGN( 50, 9) + 258 ASSIGN( 51, 9)
     + 294 ASSIGN( 52, 9) + 288 ASSIGN( 53, 9) + 342 ASSIGN( 54, 9)
     + 228 ASSIGN( 55, 9) + 204 ASSIGN( 56, 9) + 306 ASSIGN( 57, 9)
     <=    1800
END
INTE    522
```

APPENDIX C. EXAMPLE PROBLEM SOLUTION WITH M = 7

Another solution with **m** is chosen **7.**

MIN X1 + X2 + X3 + X4 + X5 + X6 + X7

SUBJECT TO

2) 5 X11 + 6 X21 + 5 X31 + 3 X41 + 2 X51 + 6 X61 <= 6

3) 5 X12 + 6 X22 + 5 X32 + 3 X42 + 2 X52 + 6 X62 <= 6

4) 5 X13 + 6 X23 + 5 X33 + 3 X43 + 2 X53 + 6 X63 <= 6

5) 5 X14 + 6 X24 + 5 X34 + 3 X44 + 2 X54 + 6 X64 <= 6

6) 5 X15 + 6 X25 + 5 X35 + 3 X45 + 2 X55 + 6 X65 <= 6

7) 5 X16 + 6 X26 + 5 X36 + 3 X46 + 2 X56 + 6 X66 <= 6

8) 5 X17 + 6 X27 + 5 X37 + 3 X47 + 2 X57 + 6 X67 <= 6

9) X11 + X12 + X13 + X14 + X15 + X16 + X17 = 1

10) X21 + X22 + X23 + X24 + X25 + X26 + X27 = 1

11) X31 + X32 + X33 + X34 + X35 + X36 + X37 = 1

12) X41 + X42 + X43 + X44 + X45 + X46 + X47 = 1

13) X51 + X52 + X53 + X54 + X55 + X56 + X57 = 1

14) X61 + X62 + X63 + X64 + X65 + X66 + X67 = 1

15) - 7 X1 + X11 + X21 + X31 + X41 + X51 + X61 <= 0

16) - 7 X2 + X12 + X22 + X32 + X42 + X52 + X62 <= 0

17) - 7 X3 + X13 + X23 + X33 + X43 + X53 + X63 <= 0

18) - 7 X4 + X14 + X24 + X34 + X44 + X54 + X64 <= 0

19) - 7 X5 + X15 + X25 + X35 + X45 + X55 + X65 <= 0

20) - 7 X6 + X16 + X26 + X36 + X46 + X56 + X66 <= 0

21) - 7 X7 + X17 + X27 + X37 + X47 + X57 + X67 <= 0

END

INTE 49

OBJECTIVE FUNCTION VALUE		
1)	5.000000	
VARIABLE	VALUE	REDUCED COST
X1	0.000000	1.000000
X2	1.000000	1.000000
X3	1.000000	1.000000
X4	0.000000	1.000000
X5	1.000000	1.000000
X6	1.000000	1.000000
X7	1.000000	1.000000
X11	0.000000	0.000000
X21	0.000000	0.000000
X31	0.000000	0.000000
X41	0.000000	0.000000
X51	0.000000	0.000000
X61	0.000000	0.000000
X12	0.000000	0.000000
X22	0.000000	0.000000
X32	0.000000	0.000000

X42	0.000000	0.000000
X52	0.000000	0.000000
X62	1.000000	0.000000
X13	1.000000	0.000000
X23	0.000000	0.000000
X33	0.000000	0.000000
X43	0.000000	0.000000
X53	0.000000	0.000000
X63	0.000000	0.000000
X14	0.000000	0.000000
X24	0.000000	0.000000
X34	0.000000	0.000000
X44	0.000000	0.000000
X54	0.000000	0.000000
X64	0.000000	0.000000
X15	0.000000	0.000000
X25	0.000000	0.000000
X35	1.000000	0.000000
X45	0.000000	0.000000
X55	0.000000	0.000000
X65	0.000000	0.000000
X16	0.000000	0.000000
X26	0.000000	0.000000
X36	0.000000	0.000000
X46	1.000000	0.000000
X56	1.000000	0.000000
X66	0.000000	0.000000
X17	0.000000	0.000000
X27	1.000000	0.000000
X37	0.000000	0.000000
X47	0.000000	0.000000

X57	0.000000	0.000000
X67	0.000000	0.000000

APPENDIX D. MISSION CHART WITH ASSOCIATED VARIABLE

MISSION NUMBER	MISSIONS	ASSOCIATED VARIABLE	MSSION TIME	
			HOURS	MINUTES
1	R1	X_{1j}	8.25	495
2	R2	X_{2j}	13.5	810
3	R3	X_{3j}	12.25	735
4	R4	X_{4j}	7.25	435
5	R5	X_{5j}	14.25	855
6	R6	X_{6j}	9.25	555
7	R7	X_{7j}	8.25	495
8	T11	X_{8j}	2.33	140
9	T12	X_{9j}	2.33	140
10	T13	X_{10j}	2.33	140
11	T14	X_{11j}	2.33	140
12	T21	X_{12j}	3.5	210
13	T22	X_{13j}	3.5	210
14	T23	X_{14j}	3.5	210
15	T24	X_{15j}	3.5	210
16	T25	X_{16j}	3.5	210
17	T26	X_{17j}	3.5	210
18	T27	X_{18j}	3.5	210
19	T31	X_{19j}	3.25	195

JOB NUMBER	MISSIONS	ASSOCIATED VARIABLE	MSSION TIME	
			HOURS	MINUTES
20	T32	X_{20j}	3.25	195
21	T33	X_{21j}	3.25	195
22	T34	X_{22j}	3.25	195
23	T35	X_{23j}	3.25	195
24	T36	X_{24j}	3.25	195
25	T37	X_{25j}	3.25	195
26	T41	X_{26j}	1.75	105
27	T42	X_{27j}	1.75	105
28	T51	X_{28j}	2.66	160
29	T52	X_{29j}	2.66	160
30	T53	X_{30j}	2.66	160
31	T54	X_{31j}	2.66	160
32	T61	X_{32j}	4.25	255
33	T62	X_{33j}	4.25	255
34	T63	X_{34j}	4.25	255
35	T64	X_{35j}	4.25	255
36	T65	X_{36j}	4.25	255
37	T66	X_{37j}	4.25	255
38	T71	X_{38j}	3.75	225
39	T72	X_{39j}	3.75	225
40	T73	X_{40j}	3.75	225

JOB NUMBER	MISSIONS	ASSOCIATED VARIABLE	MSSION TIME	
			HOURS	MINUTES
41	T74	X_{41j}	3.75	225
42	T75	X_{42j}	3.75	225
43	T81	X_{43j}	2.66	160
44	T82	X_{44j}	2.66	160
45	T83	X_{45j}	2.66	160
46	T84	X_{46j}	2.66	160
47	T85	X_{47j}	2.66	160
48	T86	X_{48j}	2.66	160
49	T87	X_{49j}	2.66	160
50	C1	X_{50j}	5.5	330
51	C2	X_{51j}	4.3	258
52	C3	X_{52j}	4.9	294
53	C4	X_{53j}	4.8	288
54	C5	X_{54j}	5.7	342
55	C6	X_{55j}	3.8	228
56	C7	X_{56j}	3.4	204
57	C8	X_{57j}	5.1	306

APPENDIX E. MIXED INTEGER SOLUTION

Global optimal solution found at step:	1305
Objective value:	9.000000
Branch count:	8

Variable	Value	Reduced Cost
AIRCRAFTNUMBER(1)	1.000000	1.000000
AIRCRAFTNUMBER(2)	1.000000	1.000000
AIRCRAFTNUMBER(3)	1.000000	1.000000
AIRCRAFTNUMBER(4)	1.000000	1.000000
AIRCRAFTNUMBER(5)	1.000000	1.000000
AIRCRAFTNUMBER(6)	1.000000	1.000000
AIRCRAFTNUMBER(7)	1.000000	1.000000
AIRCRAFTNUMBER(8)	1.000000	1.000000
AIRCRAFTNUMBER(9)	1.000000	1.000000
MISSIONLENGTH(1)	495.0000	0.0000000
MISSIONLENGTH(2)	810.0000	0.0000000
MISSIONLENGTH(3)	735.0000	0.0000000
MISSIONLENGTH(4)	435.0000	0.0000000
MISSIONLENGTH(5)	855.0000	0.0000000
MISSIONLENGTH(6)	555.0000	0.0000000
MISSIONLENGTH(7)	495.0000	0.0000000
MISSIONLENGTH(8)	140.0000	0.0000000
MISSIONLENGTH(9)	140.0000	0.0000000
MISSIONLENGTH(10)	140.0000	0.0000000
MISSIONLENGTH(11)	140.0000	0.0000000
MISSIONLENGTH(12)	210.0000	0.0000000
MISSIONLENGTH(13)	210.0000	0.0000000
MISSIONLENGTH(14)	210.0000	0.0000000
MISSIONLENGTH(15)	210.0000	0.0000000
MISSIONLENGTH(16)	210.0000	0.0000000
MISSIONLENGTH(17)	210.0000	0.0000000
MISSIONLENGTH(18)	210.0000	0.0000000
MISSIONLENGTH(19)	195.0000	0.0000000
MISSIONLENGTH(20)	195.0000	0.0000000
MISSIONLENGTH(21)	195.0000	0.0000000
MISSIONLENGTH(22)	195.0000	0.0000000
MISSIONLENGTH(23)	195.0000	0.0000000
MISSIONLENGTH(24)	195.0000	0.0000000
MISSIONLENGTH(25)	195.0000	0.0000000
MISSIONLENGTH(26)	105.0000	0.0000000

MISSIONLENGTH(27)	105.0000	0.0000000
MISSIONLENGTH(28)	160.0000	0.0000000
MISSIONLENGTH(29)	160.0000	0.0000000
MISSIONLENGTH(30)	160.0000	0.0000000
MISSIONLENGTH(31)	160.0000	0.0000000
MISSIONLENGTH(32)	255.0000	0.0000000
MISSIONLENGTH(33)	255.0000	0.0000000
MISSIONLENGTH(34)	255.0000	0.0000000
MISSIONLENGTH(35)	255.0000	0.0000000
MISSIONLENGTH(36)	255.0000	0.0000000
MISSIONLENGTH(37)	255.0000	0.0000000
MISSIONLENGTH(38)	225.0000	0.0000000
MISSIONLENGTH(39)	225.0000	0.0000000
MISSIONLENGTH(40)	225.0000	0.0000000
MISSIONLENGTH(41)	225.0000	0.0000000
MISSIONLENGTH(42)	225.0000	0.0000000
MISSIONLENGTH(43)	160.0000	0.0000000
MISSIONLENGTH(44)	160.0000	0.0000000
MISSIONLENGTH(45)	160.0000	0.0000000
MISSIONLENGTH(46)	160.0000	0.0000000
MISSIONLENGTH(47)	160.0000	0.0000000
MISSIONLENGTH(48)	160.0000	0.0000000
MISSIONLENGTH(49)	160.0000	0.0000000
MISSIONLENGTH(50)	330.0000	0.0000000
MISSIONLENGTH(51)	258.0000	0.0000000
MISSIONLENGTH(52)	294.0000	0.0000000
MISSIONLENGTH(53)	288.0000	0.0000000
MISSIONLENGTH(54)	342.0000	0.0000000
MISSIONLENGTH(55)	228.0000	0.0000000
MISSIONLENGTH(56)	204.0000	0.0000000
MISSIONLENGTH(57)	306.0000	0.0000000
ASSIGN(1, 4)	1.000000	0.0000000
ASSIGN(2, 9)	1.000000	0.0000000
ASSIGN(3, 9)	1.000000	0.0000000
ASSIGN(4, 3)	1.000000	0.0000000
ASSIGN(5, 3)	0.5380117E-01	0.0000000
ASSIGN(5, 4)	0.8771930E-01	0.0000000
ASSIGN(5, 5)	0.5602339	0.0000000
ASSIGN(5, 9)	0.2982456	0.0000000
ASSIGN(6, 6)	1.000000	0.0000000
ASSIGN(7, 6)	1.000000	0.0000000
ASSIGN(8, 5)	1.000000	0.0000000
ASSIGN(9, 5)	1.000000	0.0000000
ASSIGN(10, 6)	1.000000	0.0000000
ASSIGN(11, 1)	1.000000	0.0000000
ASSIGN(12, 3)	1.000000	0.0000000
ASSIGN(13, 6)	1.000000	0.0000000
ASSIGN(14, 3)	1.000000	0.0000000
ASSIGN(15, 1)	1.000000	0.0000000
ASSIGN(16, 4)	1.000000	0.0000000
ASSIGN(17, 4)	0.5238095	0.0000000
ASSIGN(17, 6)	0.4761905	0.0000000
ASSIGN(18, 7)	1.000000	0.0000000
ASSIGN(19, 1)	1.000000	0.0000000
ASSIGN(20, 5)	1.000000	0.0000000
ASSIGN(21, 7)	1.000000	0.0000000
ASSIGN(22, 2)	1.000000	0.0000000

ASSIGN(23, 3)	1.000000	0.0000000
ASSIGN(24, 6)	1.000000	0.0000000
ASSIGN(25, 7)	1.000000	0.0000000
ASSIGN(26, 1)	1.000000	0.0000000
ASSIGN(27, 6)	1.000000	0.0000000
ASSIGN(28, 1)	1.000000	0.0000000
ASSIGN(29, 4)	1.000000	0.0000000
ASSIGN(30, 7)	1.000000	0.0000000
ASSIGN(31, 4)	1.000000	0.0000000
ASSIGN(32, 2)	1.000000	0.0000000
ASSIGN(33, 5)	1.000000	0.0000000
ASSIGN(34, 7)	1.000000	0.0000000
ASSIGN(35, 7)	1.000000	0.0000000
ASSIGN(36, 7)	1.000000	0.0000000
ASSIGN(37, 4)	0.8039216	0.0000000
ASSIGN(37, 7)	0.1960784	0.0000000
ASSIGN(38, 1)	1.000000	0.0000000
ASSIGN(39, 1)	0.1822222	0.0000000
ASSIGN(39, 3)	0.8177778	0.0000000
ASSIGN(40, 4)	1.000000	0.0000000
ASSIGN(41, 7)	1.000000	0.0000000
ASSIGN(42, 8)	1.000000	0.0000000
ASSIGN(43, 1)	1.000000	0.0000000
ASSIGN(44, 5)	1.000000	0.0000000
ASSIGN(45, 8)	1.000000	0.0000000
ASSIGN(46, 3)	1.000000	0.0000000
ASSIGN(47, 3)	1.000000	0.0000000
ASSIGN(48, 4)	1.000000	0.0000000
ASSIGN(49, 8)	1.000000	0.0000000
ASSIGN(50, 8)	1.000000	0.0000000
ASSIGN(51, 1)	1.000000	0.0000000
ASSIGN(52, 8)	1.000000	0.0000000
ASSIGN(53, 5)	1.000000	0.0000000
ASSIGN(54, 8)	1.000000	0.0000000
ASSIGN(55, 8)	1.000000	0.0000000
ASSIGN(56, 5)	0.7009804	0.0000000
ASSIGN(56, 8)	0.2990196	0.0000000
ASSIGN(57, 1)	1.000000	0.0000000

APPENDIX F. MODIFIED FORMULATION SET

Special ordered set is replaced with the following constraint set.

```
2]  ASSIGN( 1, 1) + ASSIGN( 1, 2) + ASSIGN( 1, 3) + ASSIGN( 1, 4)
     + ASSIGN( 1, 5) + ASSIGN( 1, 6) + ASSIGN( 1, 7) + ASSIGN( 1, 8)
     + ASSIGN( 1, 9) <=   1.07
3]  ASSIGN( 2, 1) + ASSIGN( 2, 2) + ASSIGN( 2, 3) + ASSIGN( 2, 4)
     + ASSIGN( 2, 5) + ASSIGN( 2, 6) + ASSIGN( 2, 7) + ASSIGN( 2, 8)
     + ASSIGN( 2, 9) <=   1.07
4]  ASSIGN( 3, 1) + ASSIGN( 3, 2) + ASSIGN( 3, 3) + ASSIGN( 3, 4)
     + ASSIGN( 3, 5) + ASSIGN( 3, 6) + ASSIGN( 3, 7) + ASSIGN( 3, 8)
     + ASSIGN( 3, 9) <=   1.07
5]  ASSIGN( 4, 1) + ASSIGN( 4, 2) + ASSIGN( 4, 3) + ASSIGN( 4, 4)
     + ASSIGN( 4, 5) + ASSIGN( 4, 6) + ASSIGN( 4, 7) + ASSIGN( 4, 8)
     + ASSIGN( 4, 9) <=   1.07
6]  ASSIGN( 5, 1) + ASSIGN( 5, 2) + ASSIGN( 5, 3) + ASSIGN( 5, 4)
     + ASSIGN( 5, 5) + ASSIGN( 5, 6) + ASSIGN( 5, 7) + ASSIGN( 5, 8)
     + ASSIGN( 5, 9) <=   1.07
7]  ASSIGN( 6, 1) + ASSIGN( 6, 2) + ASSIGN( 6, 3) + ASSIGN( 6, 4)
     + ASSIGN( 6, 5) + ASSIGN( 6, 6) + ASSIGN( 6, 7) + ASSIGN( 6, 8)
     + ASSIGN( 6, 9) <=   1.07
8]  ASSIGN( 7, 1) + ASSIGN( 7, 2) + ASSIGN( 7, 3) + ASSIGN( 7, 4)
     + ASSIGN( 7, 5) + ASSIGN( 7, 6) + ASSIGN( 7, 7) + ASSIGN( 7, 8)
     + ASSIGN( 7, 9) <=   1.07
9]  ASSIGN( 8, 1) + ASSIGN( 8, 2) + ASSIGN( 8, 3) + ASSIGN( 8, 4)
     + ASSIGN( 8, 5) + ASSIGN( 8, 6) + ASSIGN( 8, 7) + ASSIGN( 8, 8)
     + ASSIGN( 8, 9) <=   1.07
10]  ASSIGN( 9, 1) + ASSIGN( 9, 2) + ASSIGN( 9, 3) + ASSIGN( 9, 4)
     + ASSIGN( 9, 5) + ASSIGN( 9, 6) + ASSIGN( 9, 7) + ASSIGN( 9, 8)
     + ASSIGN( 9, 9) <=   1.07
11]  ASSIGN( 10, 1) + ASSIGN( 10, 2) + ASSIGN( 10, 3) + ASSIGN( 10, 4)
     + ASSIGN( 10, 5) + ASSIGN( 10, 6) + ASSIGN( 10, 7)
     + ASSIGN( 10, 8) + ASSIGN( 10, 9) <=   1.07
12]  ASSIGN( 11, 1) + ASSIGN( 11, 2) + ASSIGN( 11, 3) + ASSIGN( 11, 4)
     + ASSIGN( 11, 5) + ASSIGN( 11, 6) + ASSIGN( 11, 7)
     + ASSIGN( 11, 8) + ASSIGN( 11, 9) <=   1.07
13]  ASSIGN( 12, 1) + ASSIGN( 12, 2) + ASSIGN( 12, 3) + ASSIGN( 12, 4)
     + ASSIGN( 12, 5) + ASSIGN( 12, 6) + ASSIGN( 12, 7)
     + ASSIGN( 12, 8) + ASSIGN( 12, 9) <=   1.07
14]  ASSIGN( 13, 1) + ASSIGN( 13, 2) + ASSIGN( 13, 3) + ASSIGN( 13, 4)
     + ASSIGN( 13, 5) + ASSIGN( 13, 6) + ASSIGN( 13, 7)
     + ASSIGN( 13, 8) + ASSIGN( 13, 9) <=   1.07
15]  ASSIGN( 14, 1) + ASSIGN( 14, 2) + ASSIGN( 14, 3) + ASSIGN( 14, 4)
     + ASSIGN( 14, 5) + ASSIGN( 14, 6) + ASSIGN( 14, 7)
     + ASSIGN( 14, 8) + ASSIGN( 14, 9) <=   1.07
16]  ASSIGN( 15, 1) + ASSIGN( 15, 2) + ASSIGN( 15, 3) + ASSIGN( 15, 4)
     + ASSIGN( 15, 5) + ASSIGN( 15, 6) + ASSIGN( 15, 7)
     + ASSIGN( 15, 8) + ASSIGN( 15, 9) <=   1.07
```

```
17] ASSIGN( 16, 1) + ASSIGN( 16, 2) + ASSIGN( 16, 3) + ASSIGN( 16, 4)
    + ASSIGN( 16, 5) + ASSIGN( 16, 6) + ASSIGN( 16, 7)
    + ASSIGN( 16, 8) + ASSIGN( 16, 9) <=   1.07
18] ASSIGN( 17, 1) + ASSIGN( 17, 2) + ASSIGN( 17, 3) + ASSIGN( 17, 4)
    + ASSIGN( 17, 5) + ASSIGN( 17, 6) + ASSIGN( 17, 7)
    + ASSIGN( 17, 8) + ASSIGN( 17, 9) <=   1.07
19] ASSIGN( 18, 1) + ASSIGN( 18, 2) + ASSIGN( 18, 3) + ASSIGN( 18, 4)
    + ASSIGN( 18, 5) + ASSIGN( 18, 6) + ASSIGN( 18, 7)
    + ASSIGN( 18, 8) + ASSIGN( 18, 9) <=   1.07
20] ASSIGN( 19, 1) + ASSIGN( 19, 2) + ASSIGN( 19, 3) + ASSIGN( 19, 4)
    + ASSIGN( 19, 5) + ASSIGN( 19, 6) + ASSIGN( 19, 7)
    + ASSIGN( 19, 8) + ASSIGN( 19, 9) <=   1.07
21] ASSIGN( 20, 1) + ASSIGN( 20, 2) + ASSIGN( 20, 3) + ASSIGN( 20, 4)
    + ASSIGN( 20, 5) + ASSIGN( 20, 6) + ASSIGN( 20, 7)
    + ASSIGN( 20, 8) + ASSIGN( 20, 9) <=   1.07
22] ASSIGN( 21, 1) + ASSIGN( 21, 2) + ASSIGN( 21, 3) + ASSIGN( 21, 4)
    + ASSIGN( 21, 5) + ASSIGN( 21, 6) + ASSIGN( 21, 7)
    + ASSIGN( 21, 8) + ASSIGN( 21, 9) <=   1.07
23] ASSIGN( 22, 1) + ASSIGN( 22, 2) + ASSIGN( 22, 3) + ASSIGN( 22, 4)
    + ASSIGN( 22, 5) + ASSIGN( 22, 6) + ASSIGN( 22, 7)
    + ASSIGN( 22, 8) + ASSIGN( 22, 9) <=   1.07
24] ASSIGN( 23, 1) + ASSIGN( 23, 2) + ASSIGN( 23, 3) + ASSIGN( 23, 4)
    + ASSIGN( 23, 5) + ASSIGN( 23, 6) + ASSIGN( 23, 7)
    + ASSIGN( 23, 8) + ASSIGN( 23, 9) <=   1.07
25] ASSIGN( 24, 1) + ASSIGN( 24, 2) + ASSIGN( 24, 3) + ASSIGN( 24, 4)
    + ASSIGN( 24, 5) + ASSIGN( 24, 6) + ASSIGN( 24, 7)
    + ASSIGN( 24, 8) + ASSIGN( 24, 9) <=   1.07
26] ASSIGN( 25, 1) + ASSIGN( 25, 2) + ASSIGN( 25, 3) + ASSIGN( 25, 4)
    + ASSIGN( 25, 5) + ASSIGN( 25, 6) + ASSIGN( 25, 7)
    + ASSIGN( 25, 8) + ASSIGN( 25, 9) <=   1.07
27] ASSIGN( 26, 1) + ASSIGN( 26, 2) + ASSIGN( 26, 3) + ASSIGN( 26, 4)
    + ASSIGN( 26, 5) + ASSIGN( 26, 6) + ASSIGN( 26, 7)
    + ASSIGN( 26, 8) + ASSIGN( 26, 9) <=   1.07
28] ASSIGN( 27, 1) + ASSIGN( 27, 2) + ASSIGN( 27, 3) + ASSIGN( 27, 4)
    + ASSIGN( 27, 5) + ASSIGN( 27, 6) + ASSIGN( 27, 7)
    + ASSIGN( 27, 8) + ASSIGN( 27, 9) <=   1.07
29] ASSIGN( 28, 1) + ASSIGN( 28, 2) + ASSIGN( 28, 3) + ASSIGN( 28, 4)
    + ASSIGN( 28, 5) + ASSIGN( 28, 6) + ASSIGN( 28, 7)
    + ASSIGN( 28, 8) + ASSIGN( 28, 9) <=   1.07
30] ASSIGN( 29, 1) + ASSIGN( 29, 2) + ASSIGN( 29, 3) + ASSIGN( 29, 4)
    + ASSIGN( 29, 5) + ASSIGN( 29, 6) + ASSIGN( 29, 7)
    + ASSIGN( 29, 8) + ASSIGN( 29, 9) <=   1.07
31] ASSIGN( 30, 1) + ASSIGN( 30, 2) + ASSIGN( 30, 3) + ASSIGN( 30, 4)
    + ASSIGN( 30, 5) + ASSIGN( 30, 6) + ASSIGN( 30, 7)
    + ASSIGN( 30, 8) + ASSIGN( 30, 9) <=   1.07
32] ASSIGN( 31, 1) + ASSIGN( 31, 2) + ASSIGN( 31, 3) + ASSIGN( 31, 4)
    + ASSIGN( 31, 5) + ASSIGN( 31, 6) + ASSIGN( 31, 7)
    + ASSIGN( 31, 8) + ASSIGN( 31, 9) <=   1.07
33] ASSIGN( 32, 1) + ASSIGN( 32, 2) + ASSIGN( 32, 3) + ASSIGN( 32, 4)
    + ASSIGN( 32, 5) + ASSIGN( 32, 6) + ASSIGN( 32, 7)
    + ASSIGN( 32, 8) + ASSIGN( 32, 9) <=   1.07
34] ASSIGN( 33, 1) + ASSIGN( 33, 2) + ASSIGN( 33, 3) + ASSIGN( 33, 4)
    + ASSIGN( 33, 5) + ASSIGN( 33, 6) + ASSIGN( 33, 7)
    + ASSIGN( 33, 8) + ASSIGN( 33, 9) <=   1.07
35] ASSIGN( 34, 1) + ASSIGN( 34, 2) + ASSIGN( 34, 3) + ASSIGN( 34, 4)
    + ASSIGN( 34, 5) + ASSIGN( 34, 6) + ASSIGN( 34, 7)
    + ASSIGN( 34, 8) + ASSIGN( 34, 9) <=   1.07
```

```
36]  ASSIGN( 35, 1) + ASSIGN( 35, 2) + ASSIGN( 35, 3) + ASSIGN( 35, 4)
     + ASSIGN( 35, 5) + ASSIGN( 35, 6) + ASSIGN( 35, 7)
     + ASSIGN( 35, 8) + ASSIGN( 35, 9) <=   1.07
37]  ASSIGN( 36, 1) + ASSIGN( 36, 2) + ASSIGN( 36, 3) + ASSIGN( 36, 4)
     + ASSIGN( 36, 5) + ASSIGN( 36, 6) + ASSIGN( 36, 7)
     + ASSIGN( 36, 8) + ASSIGN( 36, 9) <=   1.07
38]  ASSIGN( 37, 1) + ASSIGN( 37, 2) + ASSIGN( 37, 3) + ASSIGN( 37, 4)
     + ASSIGN( 37, 5) + ASSIGN( 37, 6) + ASSIGN( 37, 7)
     + ASSIGN( 37, 8) + ASSIGN( 37, 9) <=   1.07
39]  ASSIGN( 38, 1) + ASSIGN( 38, 2) + ASSIGN( 38, 3) + ASSIGN( 38, 4)
     + ASSIGN( 38, 5) + ASSIGN( 38, 6) + ASSIGN( 38, 7)
     + ASSIGN( 38, 8) + ASSIGN( 38, 9) <=   1.07
40]  ASSIGN( 39, 1) + ASSIGN( 39, 2) + ASSIGN( 39, 3) + ASSIGN( 39, 4)
     + ASSIGN( 39, 5) + ASSIGN( 39, 6) + ASSIGN( 39, 7)
     + ASSIGN( 39, 8) + ASSIGN( 39, 9) <=   1.07
41]  ASSIGN( 40, 1) + ASSIGN( 40, 2) + ASSIGN( 40, 3) + ASSIGN( 40, 4)
     + ASSIGN( 40, 5) + ASSIGN( 40, 6) + ASSIGN( 40, 7)
     + ASSIGN( 40, 8) + ASSIGN( 40, 9) <=   1.07
42]  ASSIGN( 41, 1) + ASSIGN( 41, 2) + ASSIGN( 41, 3) + ASSIGN( 41, 4)
     + ASSIGN( 41, 5) + ASSIGN( 41, 6) + ASSIGN( 41, 7)
     + ASSIGN( 41, 8) + ASSIGN( 41, 9) <=   1.07
43]  ASSIGN( 42, 1) + ASSIGN( 42, 2) + ASSIGN( 42, 3) + ASSIGN( 42, 4)
     + ASSIGN( 42, 5) + ASSIGN( 42, 6) + ASSIGN( 42, 7)
     + ASSIGN( 42, 8) + ASSIGN( 42, 9) <=   1.07
44]  ASSIGN( 43, 1) + ASSIGN( 43, 2) + ASSIGN( 43, 3) + ASSIGN( 43, 4)
     + ASSIGN( 43, 5) + ASSIGN( 43, 6) + ASSIGN( 43, 7)
     + ASSIGN( 43, 8) + ASSIGN( 43, 9) <=   1.07
45]  ASSIGN( 44, 1) + ASSIGN( 44, 2) + ASSIGN( 44, 3) + ASSIGN( 44, 4)
     + ASSIGN( 44, 5) + ASSIGN( 44, 6) + ASSIGN( 44, 7)
     + ASSIGN( 44, 8) + ASSIGN( 44, 9) <=   1.07
46]  ASSIGN( 45, 1) + ASSIGN( 45, 2) + ASSIGN( 45, 3) + ASSIGN( 45, 4)
     + ASSIGN( 45, 5) + ASSIGN( 45, 6) + ASSIGN( 45, 7)
     + ASSIGN( 45, 8) + ASSIGN( 45, 9) <=   1.07
47]  ASSIGN( 46, 1) + ASSIGN( 46, 2) + ASSIGN( 46, 3) + ASSIGN( 46, 4)
     + ASSIGN( 46, 5) + ASSIGN( 46, 6) + ASSIGN( 46, 7)
     + ASSIGN( 46, 8) + ASSIGN( 46, 9) <=   1.07
48]  ASSIGN( 47, 1) + ASSIGN( 47, 2) + ASSIGN( 47, 3) + ASSIGN( 47, 4)
     + ASSIGN( 47, 5) + ASSIGN( 47, 6) + ASSIGN( 47, 7)
     + ASSIGN( 47, 8) + ASSIGN( 47, 9) <=   1.07
49]  ASSIGN( 48, 1) + ASSIGN( 48, 2) + ASSIGN( 48, 3) + ASSIGN( 48, 4)
     + ASSIGN( 48, 5) + ASSIGN( 48, 6) + ASSIGN( 48, 7)
     + ASSIGN( 48, 8) + ASSIGN( 48, 9) <=   1.07
50]  ASSIGN( 49, 1) + ASSIGN( 49, 2) + ASSIGN( 49, 3) + ASSIGN( 49, 4)
     + ASSIGN( 49, 5) + ASSIGN( 49, 6) + ASSIGN( 49, 7)
     + ASSIGN( 49, 8) + ASSIGN( 49, 9) <=   1.07
51]  ASSIGN( 50, 1) + ASSIGN( 50, 2) + ASSIGN( 50, 3) + ASSIGN( 50, 4)
     + ASSIGN( 50, 5) + ASSIGN( 50, 6) + ASSIGN( 50, 7)
     + ASSIGN( 50, 8) + ASSIGN( 50, 9) <=   1.07
52]  ASSIGN( 51, 1) + ASSIGN( 51, 2) + ASSIGN( 51, 3) + ASSIGN( 51, 4)
     + ASSIGN( 51, 5) + ASSIGN( 51, 6) + ASSIGN( 51, 7)
     + ASSIGN( 51, 8) + ASSIGN( 51, 9) <=   1.07
53]  ASSIGN( 52, 1) + ASSIGN( 52, 2) + ASSIGN( 52, 3) + ASSIGN( 52, 4)
     + ASSIGN( 52, 5) + ASSIGN( 52, 6) + ASSIGN( 52, 7)
     + ASSIGN( 52, 8) + ASSIGN( 52, 9) <=   1.07
54]  ASSIGN( 53, 1) + ASSIGN( 53, 2) + ASSIGN( 53, 3) + ASSIGN( 53, 4)
     + ASSIGN( 53, 5) + ASSIGN( 53, 6) + ASSIGN( 53, 7)
     + ASSIGN( 53, 8) + ASSIGN( 53, 9) <=   1.07
```

```
55]  ASSIGN( 54, 1) + ASSIGN( 54, 2) + ASSIGN( 54, 3) + ASSIGN( 54, 4)
     + ASSIGN( 54, 5) + ASSIGN( 54, 6) + ASSIGN( 54, 7)
     + ASSIGN( 54, 8) + ASSIGN( 54, 9) <=   1.07
56]  ASSIGN( 55, 1) + ASSIGN( 55, 2) + ASSIGN( 55, 3) + ASSIGN( 55, 4)
     + ASSIGN( 55, 5) + ASSIGN( 55, 6) + ASSIGN( 55, 7)
     + ASSIGN( 55, 8) + ASSIGN( 55, 9) <=   1.07
57]  ASSIGN( 56, 1) + ASSIGN( 56, 2) + ASSIGN( 56, 3) + ASSIGN( 56, 4)
     + ASSIGN( 56, 5) + ASSIGN( 56, 6) + ASSIGN( 56, 7)
     + ASSIGN( 56, 8) + ASSIGN( 56, 9) <=   1.07
58]  ASSIGN( 57, 1) + ASSIGN( 57, 2) + ASSIGN( 57, 3) + ASSIGN( 57, 4)
     + ASSIGN( 57, 5) + ASSIGN( 57, 6) + ASSIGN( 57, 7)
     + ASSIGN( 57, 8) + ASSIGN( 57, 9) <=   1.07
59]  ASSIGN( 1, 1) + ASSIGN( 1, 2) + ASSIGN( 1, 3) + ASSIGN( 1, 4)
     + ASSIGN( 1, 5) + ASSIGN( 1, 6) + ASSIGN( 1, 7) + ASSIGN( 1, 8)
     + ASSIGN( 1, 9) >=   .93
60]  ASSIGN( 2, 1) + ASSIGN( 2, 2) + ASSIGN( 2, 3) + ASSIGN( 2, 4)
     + ASSIGN( 2, 5) + ASSIGN( 2, 6) + ASSIGN( 2, 7) + ASSIGN( 2, 8)
     + ASSIGN( 2, 9) >=   .93
61]  ASSIGN( 3, 1) + ASSIGN( 3, 2) + ASSIGN( 3, 3) + ASSIGN( 3, 4)
     + ASSIGN( 3, 5) + ASSIGN( 3, 6) + ASSIGN( 3, 7) + ASSIGN( 3, 8)
     + ASSIGN( 3, 9) >=   .93
62]  ASSIGN( 4, 1) + ASSIGN( 4, 2) + ASSIGN( 4, 3) + ASSIGN( 4, 4)
     + ASSIGN( 4, 5) + ASSIGN( 4, 6) + ASSIGN( 4, 7) + ASSIGN( 4, 8)
     + ASSIGN( 4, 9) >=   .93
63]  ASSIGN( 5, 1) + ASSIGN( 5, 2) + ASSIGN( 5, 3) + ASSIGN( 5, 4)
     + ASSIGN( 5, 5) + ASSIGN( 5, 6) + ASSIGN( 5, 7) + ASSIGN( 5, 8)
     + ASSIGN( 5, 9) >=   .93
64]  ASSIGN( 6, 1) + ASSIGN( 6, 2) + ASSIGN( 6, 3) + ASSIGN( 6, 4)
     + ASSIGN( 6, 5) + ASSIGN( 6, 6) + ASSIGN( 6, 7) + ASSIGN( 6, 8)
     + ASSIGN( 6, 9) >=   .93
65]  ASSIGN( 7, 1) + ASSIGN( 7, 2) + ASSIGN( 7, 3) + ASSIGN( 7, 4)
     + ASSIGN( 7, 5) + ASSIGN( 7, 6) + ASSIGN( 7, 7) + ASSIGN( 7, 8)
     + ASSIGN( 7, 9) >=   .93
66]  ASSIGN( 8, 1) + ASSIGN( 8, 2) + ASSIGN( 8, 3) + ASSIGN( 8, 4)
     + ASSIGN( 8, 5) + ASSIGN( 8, 6) + ASSIGN( 8, 7) + ASSIGN( 8, 8)
     + ASSIGN( 8, 9) >=   .93
67]  ASSIGN( 9, 1) + ASSIGN( 9, 2) + ASSIGN( 9, 3) + ASSIGN( 9, 4)
     + ASSIGN( 9, 5) + ASSIGN( 9, 6) + ASSIGN( 9, 7) + ASSIGN( 9, 8)
     + ASSIGN( 9, 9) >=   .93
68]  ASSIGN( 10, 1) + ASSIGN( 10, 2) + ASSIGN( 10, 3) + ASSIGN( 10, 4)
     + ASSIGN( 10, 5) + ASSIGN( 10, 6) + ASSIGN( 10, 7)
     + ASSIGN( 10, 8) + ASSIGN( 10, 9) >=   .93
69]  ASSIGN( 11, 1) + ASSIGN( 11, 2) + ASSIGN( 11, 3) + ASSIGN( 11, 4)
     + ASSIGN( 11, 5) + ASSIGN( 11, 6) + ASSIGN( 11, 7)
     + ASSIGN( 11, 8) + ASSIGN( 11, 9) >=   .93
70]  ASSIGN( 12, 1) + ASSIGN( 12, 2) + ASSIGN( 12, 3) + ASSIGN( 12, 4)
     + ASSIGN( 12, 5) + ASSIGN( 12, 6) + ASSIGN( 12, 7)
     + ASSIGN( 12, 8) + ASSIGN( 12, 9) >=   .93
71]  ASSIGN( 13, 1) + ASSIGN( 13, 2) + ASSIGN( 13, 3) + ASSIGN( 13, 4)
     + ASSIGN( 13, 5) + ASSIGN( 13, 6) + ASSIGN( 13, 7)
     + ASSIGN( 13, 8) + ASSIGN( 13, 9) >=   .93
72]  ASSIGN( 14, 1) + ASSIGN( 14, 2) + ASSIGN( 14, 3) + ASSIGN( 14, 4)
     + ASSIGN( 14, 5) + ASSIGN( 14, 6) + ASSIGN( 14, 7)
     + ASSIGN( 14, 8) + ASSIGN( 14, 9) >=   .93
73]  ASSIGN( 15, 1) + ASSIGN( 15, 2) + ASSIGN( 15, 3) + ASSIGN( 15, 4)
     + ASSIGN( 15, 5) + ASSIGN( 15, 6) + ASSIGN( 15, 7)
     + ASSIGN( 15, 8) + ASSIGN( 15, 9) >=   .93
```

```
74]  ASSIGN( 16, 1) + ASSIGN( 16, 2) + ASSIGN( 16, 3) + ASSIGN( 16, 4)
     + ASSIGN( 16, 5) + ASSIGN( 16, 6) + ASSIGN( 16, 7)
     + ASSIGN( 16, 8) + ASSIGN( 16, 9) >=   .93
75]  ASSIGN( 17, 1) + ASSIGN( 17, 2) + ASSIGN( 17, 3) + ASSIGN( 17, 4)
     + ASSIGN( 17, 5) + ASSIGN( 17, 6) + ASSIGN( 17, 7)
     + ASSIGN( 17, 8) + ASSIGN( 17, 9) >=   .93
76]  ASSIGN( 18, 1) + ASSIGN( 18, 2) + ASSIGN( 18, 3) + ASSIGN( 18, 4)
     + ASSIGN( 18, 5) + ASSIGN( 18, 6) + ASSIGN( 18, 7)
     + ASSIGN( 18, 8) + ASSIGN( 18, 9) >=   .93
77]  ASSIGN( 19, 1) + ASSIGN( 19, 2) + ASSIGN( 19, 3) + ASSIGN( 19, 4)
     + ASSIGN( 19, 5) + ASSIGN( 19, 6) + ASSIGN( 19, 7)
     + ASSIGN( 19, 8) + ASSIGN( 19, 9) >=   .93
78]  ASSIGN( 20, 1) + ASSIGN( 20, 2) + ASSIGN( 20, 3) + ASSIGN( 20, 4)
     + ASSIGN( 20, 5) + ASSIGN( 20, 6) + ASSIGN( 20, 7)
     + ASSIGN( 20, 8) + ASSIGN( 20, 9) >=   .93
79]  ASSIGN( 21, 1) + ASSIGN( 21, 2) + ASSIGN( 21, 3) + ASSIGN( 21, 4)
     + ASSIGN( 21, 5) + ASSIGN( 21, 6) + ASSIGN( 21, 7)
     + ASSIGN( 21, 8) + ASSIGN( 21, 9) >=   .93
80]  ASSIGN( 22, 1) + ASSIGN( 22, 2) + ASSIGN( 22, 3) + ASSIGN( 22, 4)
     + ASSIGN( 22, 5) + ASSIGN( 22, 6) + ASSIGN( 22, 7)
     + ASSIGN( 22, 8) + ASSIGN( 22, 9) >=   .93
81]  ASSIGN( 23, 1) + ASSIGN( 23, 2) + ASSIGN( 23, 3) + ASSIGN( 23, 4)
     + ASSIGN( 23, 5) + ASSIGN( 23, 6) + ASSIGN( 23, 7)
     + ASSIGN( 23, 8) + ASSIGN( 23, 9) >=   .93
82]  ASSIGN( 24, 1) + ASSIGN( 24, 2) + ASSIGN( 24, 3) + ASSIGN( 24, 4)
     + ASSIGN( 24, 5) + ASSIGN( 24, 6) + ASSIGN( 24, 7)
     + ASSIGN( 24, 8) + ASSIGN( 24, 9) >=   .93
83]  ASSIGN( 25, 1) + ASSIGN( 25, 2) + ASSIGN( 25, 3) + ASSIGN( 25, 4)
     + ASSIGN( 25, 5) + ASSIGN( 25, 6) + ASSIGN( 25, 7)
     + ASSIGN( 25, 8) + ASSIGN( 25, 9) >=   .93
84]  ASSIGN( 26, 1) + ASSIGN( 26, 2) + ASSIGN( 26, 3) + ASSIGN( 26, 4)
     + ASSIGN( 26, 5) + ASSIGN( 26, 6) + ASSIGN( 26, 7)
     + ASSIGN( 26, 8) + ASSIGN( 26, 9) >=   .93
85]  ASSIGN( 27, 1) + ASSIGN( 27, 2) + ASSIGN( 27, 3) + ASSIGN( 27, 4)
     + ASSIGN( 27, 5) + ASSIGN( 27, 6) + ASSIGN( 27, 7)
     + ASSIGN( 27, 8) + ASSIGN( 27, 9) >=   .93
86]  ASSIGN( 28, 1) + ASSIGN( 28, 2) + ASSIGN( 28, 3) + ASSIGN( 28, 4)
     + ASSIGN( 28, 5) + ASSIGN( 28, 6) + ASSIGN( 28, 7)
     + ASSIGN( 28, 8) + ASSIGN( 28, 9) >=   .93
87]  ASSIGN( 29, 1) + ASSIGN( 29, 2) + ASSIGN( 29, 3) + ASSIGN( 29, 4)
     + ASSIGN( 29, 5) + ASSIGN( 29, 6) + ASSIGN( 29, 7)
     + ASSIGN( 29, 8) + ASSIGN( 29, 9) >=   .93
88]  ASSIGN( 30, 1) + ASSIGN( 30, 2) + ASSIGN( 30, 3) + ASSIGN( 30, 4)
     + ASSIGN( 30, 5) + ASSIGN( 30, 6) + ASSIGN( 30, 7)
     + ASSIGN( 30, 8) + ASSIGN( 30, 9) >=   .93
89]  ASSIGN( 31, 1) + ASSIGN( 31, 2) + ASSIGN( 31, 3) + ASSIGN( 31, 4)
     + ASSIGN( 31, 5) + ASSIGN( 31, 6) + ASSIGN( 31, 7)
     + ASSIGN( 31, 8) + ASSIGN( 31, 9) >=   .93
90]  ASSIGN( 32, 1) + ASSIGN( 32, 2) + ASSIGN( 32, 3) + ASSIGN( 32, 4)
     + ASSIGN( 32, 5) + ASSIGN( 32, 6) + ASSIGN( 32, 7)
     + ASSIGN( 32, 8) + ASSIGN( 32, 9) >=   .93
91]  ASSIGN( 33, 1) + ASSIGN( 33, 2) + ASSIGN( 33, 3) + ASSIGN( 33, 4)
     + ASSIGN( 33, 5) + ASSIGN( 33, 6) + ASSIGN( 33, 7)
     + ASSIGN( 33, 8) + ASSIGN( 33, 9) >=   .93
92]  ASSIGN( 34, 1) + ASSIGN( 34, 2) + ASSIGN( 34, 3) + ASSIGN( 34, 4)
     + ASSIGN( 34, 5) + ASSIGN( 34, 6) + ASSIGN( 34, 7)
     + ASSIGN( 34, 8) + ASSIGN( 34, 9) >=   .93
```

```
93]  ASSIGN( 35, 1) + ASSIGN( 35, 2) + ASSIGN( 35, 3) + ASSIGN( 35, 4)
     + ASSIGN( 35, 5) + ASSIGN( 35, 6) + ASSIGN( 35, 7)
     + ASSIGN( 35, 8) + ASSIGN( 35, 9) >=   .93
94]  ASSIGN( 36, 1) + ASSIGN( 36, 2) + ASSIGN( 36, 3) + ASSIGN( 36, 4)
     + ASSIGN( 36, 5) + ASSIGN( 36, 6) + ASSIGN( 36, 7)
     + ASSIGN( 36, 8) + ASSIGN( 36, 9) >=   .93
95]  ASSIGN( 37, 1) + ASSIGN( 37, 2) + ASSIGN( 37, 3) + ASSIGN( 37, 4)
     + ASSIGN( 37, 5) + ASSIGN( 37, 6) + ASSIGN( 37, 7)
     + ASSIGN( 37, 8) + ASSIGN( 37, 9) >=   .93
96]  ASSIGN( 38, 1) + ASSIGN( 38, 2) + ASSIGN( 38, 3) + ASSIGN( 38, 4)
     + ASSIGN( 38, 5) + ASSIGN( 38, 6) + ASSIGN( 38, 7)
     + ASSIGN( 38, 8) + ASSIGN( 38, 9) >=   .93
97]  ASSIGN( 39, 1) + ASSIGN( 39, 2) + ASSIGN( 39, 3) + ASSIGN( 39, 4)
     + ASSIGN( 39, 5) + ASSIGN( 39, 6) + ASSIGN( 39, 7)
     + ASSIGN( 39, 8) + ASSIGN( 39, 9) >=   .93
98]  ASSIGN( 40, 1) + ASSIGN( 40, 2) + ASSIGN( 40, 3) + ASSIGN( 40, 4)
     + ASSIGN( 40, 5) + ASSIGN( 40, 6) + ASSIGN( 40, 7)
     + ASSIGN( 40, 8) + ASSIGN( 40, 9) >=   .93
99]  ASSIGN( 41, 1) + ASSIGN( 41, 2) + ASSIGN( 41, 3) + ASSIGN( 41, 4)
     + ASSIGN( 41, 5) + ASSIGN( 41, 6) + ASSIGN( 41, 7)
     + ASSIGN( 41, 8) + ASSIGN( 41, 9) >=   .93
100]  ASSIGN( 42, 1) + ASSIGN( 42, 2) + ASSIGN( 42, 3)
     + ASSIGN( 42, 4) + ASSIGN( 42, 5) + ASSIGN( 42, 6)
     + ASSIGN( 42, 7) + ASSIGN( 42, 8) + ASSIGN( 42, 9) >=    .93
101]  ASSIGN( 43, 1) + ASSIGN( 43, 2) + ASSIGN( 43, 3)
     + ASSIGN( 43, 4) + ASSIGN( 43, 5) + ASSIGN( 43, 6)
     + ASSIGN( 43, 7) + ASSIGN( 43, 8) + ASSIGN( 43, 9) >=    .93
102]  ASSIGN( 44, 1) + ASSIGN( 44, 2) + ASSIGN( 44, 3)
     + ASSIGN( 44, 4) + ASSIGN( 44, 5) + ASSIGN( 44, 6)
     + ASSIGN( 44, 7) + ASSIGN( 44, 8) + ASSIGN( 44, 9) >=    .93
103]  ASSIGN( 45, 1) + ASSIGN( 45, 2) + ASSIGN( 45, 3)
     + ASSIGN( 45, 4) + ASSIGN( 45, 5) + ASSIGN( 45, 6)
     + ASSIGN( 45, 7) + ASSIGN( 45, 8) + ASSIGN( 45, 9) >=    .93
104]  ASSIGN( 46, 1) + ASSIGN( 46, 2) + ASSIGN( 46, 3)
     + ASSIGN( 46, 4) + ASSIGN( 46, 5) + ASSIGN( 46, 6)
     + ASSIGN( 46, 7) + ASSIGN( 46, 8) + ASSIGN( 46, 9) >=    .93
105]  ASSIGN( 47, 1) + ASSIGN( 47, 2) + ASSIGN( 47, 3)
     + ASSIGN( 47, 4) + ASSIGN( 47, 5) + ASSIGN( 47, 6)
     + ASSIGN( 47, 7) + ASSIGN( 47, 8) + ASSIGN( 47, 9) >=    .93
106]  ASSIGN( 48, 1) + ASSIGN( 48, 2) + ASSIGN( 48, 3)
     + ASSIGN( 48, 4) + ASSIGN( 48, 5) + ASSIGN( 48, 6)
     + ASSIGN( 48, 7) + ASSIGN( 48, 8) + ASSIGN( 48, 9) >=    .93
107]  ASSIGN( 49, 1) + ASSIGN( 49, 2) + ASSIGN( 49, 3)
     + ASSIGN( 49, 4) + ASSIGN( 49, 5) + ASSIGN( 49, 6)
     + ASSIGN( 49, 7) + ASSIGN( 49, 8) + ASSIGN( 49, 9) >=    .93
108]  ASSIGN( 50, 1) + ASSIGN( 50, 2) + ASSIGN( 50, 3)
     + ASSIGN( 50, 4) + ASSIGN( 50, 5) + ASSIGN( 50, 6)
     + ASSIGN( 50, 7) + ASSIGN( 50, 8) + ASSIGN( 50, 9) >=    .93
109]  ASSIGN( 51, 1) + ASSIGN( 51, 2) + ASSIGN( 51, 3)
     + ASSIGN( 51, 4) + ASSIGN( 51, 5) + ASSIGN( 51, 6)
     + ASSIGN( 51, 7) + ASSIGN( 51, 8) + ASSIGN( 51, 9) >=    .93
110]  ASSIGN( 52, 1) + ASSIGN( 52, 2) + ASSIGN( 52, 3)
     + ASSIGN( 52, 4) + ASSIGN( 52, 5) + ASSIGN( 52, 6)
     + ASSIGN( 52, 7) + ASSIGN( 52, 8) + ASSIGN( 52, 9) >=    .93
111]  ASSIGN( 53, 1) + ASSIGN( 53, 2) + ASSIGN( 53, 3)
     + ASSIGN( 53, 4) + ASSIGN( 53, 5) + ASSIGN( 53, 6)
     + ASSIGN( 53, 7) + ASSIGN( 53, 8) + ASSIGN( 53, 9) >=    .93
```

```
112]  ASSIGN( 54, 1) + ASSIGN( 54, 2) + ASSIGN( 54, 3)
     + ASSIGN( 54, 4) + ASSIGN( 54, 5) + ASSIGN( 54, 6)
     + ASSIGN( 54, 7) + ASSIGN( 54, 8) + ASSIGN( 54, 9) >=    .93
113]  ASSIGN( 55, 1) + ASSIGN( 55, 2) + ASSIGN( 55, 3)
     + ASSIGN( 55, 4) + ASSIGN( 55, 5) + ASSIGN( 55, 6)
     + ASSIGN( 55, 7) + ASSIGN( 55, 8) + ASSIGN( 55, 9) >=    .93
114]  ASSIGN( 56, 1) + ASSIGN( 56, 2) + ASSIGN( 56, 3)
     + ASSIGN( 56, 4) + ASSIGN( 56, 5) + ASSIGN( 56, 6)
     + ASSIGN( 56, 7) + ASSIGN( 56, 8) + ASSIGN( 56, 9) >=    .93
115]  ASSIGN( 57, 1) + ASSIGN( 57, 2) + ASSIGN( 57, 3)
     + ASSIGN( 57, 4) + ASSIGN( 57, 5) + ASSIGN( 57, 6)
     + ASSIGN( 57, 7) + ASSIGN( 57, 8) + ASSIGN( 57, 9) >=    .93
```

APPENDIX G. SOLUTION OUTPUT

Solution (With Upper Bound of Missions)

Global optimal solution found at step:	**134431935**
Objective value:	**9.000000**
Branch count:	**750980**

Variable	Value	Reduced Cost
AIRCRAFTNUMBER(1)	1.000000	1.000000
AIRCRAFTNUMBER(2)	1.000000	1.000000
AIRCRAFTNUMBER(3)	1.000000	1.000000
AIRCRAFTNUMBER(4)	1.000000	1.000000
AIRCRAFTNUMBER(5)	1.000000	1.000000
AIRCRAFTNUMBER(6)	1.000000	1.000000
AIRCRAFTNUMBER(7)	1.000000	1.000000
AIRCRAFTNUMBER(8)	1.000000	1.000000
AIRCRAFTNUMBER(9)	1.000000	1.000000
MISSIONLENGTH(1)	495.0000	0.0000000
MISSIONLENGTH(2)	810.0000	0.0000000
MISSIONLENGTH(3)	735.0000	0.0000000
MISSIONLENGTH(4)	435.0000	0.0000000
MISSIONLENGTH(5)	855.0000	0.0000000
MISSIONLENGTH(6)	555.0000	0.0000000
MISSIONLENGTH(7)	495.0000	0.0000000
MISSIONLENGTH(8)	140.0000	0.0000000
MISSIONLENGTH(9)	140.0000	0.0000000
MISSIONLENGTH(10)	140.0000	0.0000000
MISSIONLENGTH(11)	140.0000	0.0000000
MISSIONLENGTH(12)	210.0000	0.0000000
MISSIONLENGTH(13)	210.0000	0.0000000
MISSIONLENGTH(14)	210.0000	0.0000000
MISSIONLENGTH(15)	210.0000	0.0000000
MISSIONLENGTH(16)	210.0000	0.0000000
MISSIONLENGTH(17)	210.0000	0.0000000
MISSIONLENGTH(18)	210.0000	0.0000000
MISSIONLENGTH(19)	195.0000	0.0000000
MISSIONLENGTH(20)	195.0000	0.0000000
MISSIONLENGTH(21)	195.0000	0.0000000

MISSIONLENGTH(22)	195.0000	0.0000000
MISSIONLENGTH(23)	195.0000	0.0000000
MISSIONLENGTH(24)	195.0000	0.0000000
MISSIONLENGTH(25)	195.0000	0.0000000
MISSIONLENGTH(26)	105.0000	0.0000000
MISSIONLENGTH(27)	105.0000	0.0000000
MISSIONLENGTH(28)	160.0000	0.0000000
MISSIONLENGTH(29)	160.0000	0.0000000
MISSIONLENGTH(30)	160.0000	0.0000000
MISSIONLENGTH(31)	160.0000	0.0000000
MISSIONLENGTH(32)	255.0000	0.0000000
MISSIONLENGTH(33)	255.0000	0.0000000
MISSIONLENGTH(34)	255.0000	0.0000000
MISSIONLENGTH(35)	255.0000	0.0000000
MISSIONLENGTH(36)	255.0000	0.0000000
MISSIONLENGTH(37)	255.0000	0.0000000
MISSIONLENGTH(38)	225.0000	0.0000000
MISSIONLENGTH(39)	225.0000	0.0000000
MISSIONLENGTH(40)	225.0000	0.0000000
MISSIONLENGTH(41)	225.0000	0.0000000
MISSIONLENGTH(42)	225.0000	0.0000000
MISSIONLENGTH(43)	160.0000	0.0000000
MISSIONLENGTH(44)	160.0000	0.0000000
MISSIONLENGTH(45)	160.0000	0.0000000
MISSIONLENGTH(46)	160.0000	0.0000000
MISSIONLENGTH(47)	160.0000	0.0000000
MISSIONLENGTH(48)	160.0000	0.0000000
MISSIONLENGTH(49)	160.0000	0.0000000
MISSIONLENGTH(50)	330.0000	0.0000000
MISSIONLENGTH(51)	258.0000	0.0000000
MISSIONLENGTH(52)	294.0000	0.0000000
MISSIONLENGTH(53)	288.0000	0.0000000
MISSIONLENGTH(54)	342.0000	0.0000000
MISSIONLENGTH(55)	228.0000	0.0000000
MISSIONLENGTH(56)	204.0000	0.0000000
MISSIONLENGTH(57)	306.0000	0.0000000
ASSIGN(1, 2)	1.000000	0.0000000
ASSIGN(2, 4)	1.000000	0.0000000
ASSIGN(3, 9)	1.000000	0.0000000
ASSIGN(4, 1)	1.000000	0.0000000
ASSIGN(5, 3)	1.000000	0.0000000
ASSIGN(6, 5)	1.000000	0.0000000
ASSIGN(7, 1)	1.000000	0.0000000
ASSIGN(8, 8)	1.000000	0.0000000
ASSIGN(9, 7)	1.000000	0.0000000
ASSIGN(10, 7)	1.000000	0.0000000
ASSIGN(11, 9)	1.000000	0.0000000
ASSIGN(12, 6)	1.000000	0.0000000
ASSIGN(13, 3)	1.000000	0.0000000
ASSIGN(14, 2)	1.000000	0.0000000
ASSIGN(15, 2)	1.000000	0.0000000
ASSIGN(16, 2)	1.000000	0.0000000
ASSIGN(17, 2)	1.000000	0.0000000
ASSIGN(18, 5)	1.000000	0.0000000
ASSIGN(19, 6)	1.000000	0.0000000
ASSIGN(20, 8)	1.000000	0.0000000
ASSIGN(21, 6)	1.000000	0.0000000

ASSIGN(22, 9)	1.000000	0.0000000
ASSIGN(23, 6)	1.000000	0.0000000
ASSIGN(24, 6)	1.000000	0.0000000
ASSIGN(25, 4)	1.000000	0.0000000
ASSIGN(26, 9)	1.000000	0.0000000
ASSIGN(27, 9)	1.000000	0.0000000
ASSIGN(28, 7)	1.000000	0.0000000
ASSIGN(29, 2)	1.000000	0.0000000
ASSIGN(30, 8)	1.000000	0.0000000
ASSIGN(31, 8)	1.000000	0.0000000
ASSIGN(32, 6)	1.000000	0.0000000
ASSIGN(33, 1)	1.000000	0.0000000
ASSIGN(34, 6)	1.000000	0.0000000
ASSIGN(35, 8)	1.000000	0.0000000
ASSIGN(36, 8)	1.000000	0.0000000
ASSIGN(37, 3)	1.000000	0.0000000
ASSIGN(38, 8)	1.000000	0.0000000
ASSIGN(39, 8)	1.000000	0.0000000
ASSIGN(40, 9)	1.000000	0.0000000
ASSIGN(41, 6)	1.000000	0.0000000
ASSIGN(42, 2)	1.000000	0.0000000
ASSIGN(43, 9)	1.000000	0.0000000
ASSIGN(44, 8)	1.000000	0.0000000
ASSIGN(45, 7)	1.000000	0.0000000
ASSIGN(46, 7)	1.000000	0.0000000
ASSIGN(47, 3)	1.000000	0.0000000
ASSIGN(48, 3)	1.000000	0.0000000
ASSIGN(49, 3)	1.000000	0.0000000
ASSIGN(50, 7)	1.000000	0.0000000
ASSIGN(51, 4)	1.000000	0.0000000
ASSIGN(52, 4)	1.000000	0.0000000
ASSIGN(53, 5)	1.000000	0.0000000
ASSIGN(54, 5)	1.000000	0.0000000
ASSIGN(55, 4)	1.000000	0.0000000
ASSIGN(56, 1)	1.000000	0.0000000
ASSIGN(57, 5)	1.000000	0.0000000

Solution (With Lower Bound of Mission)

Global optimal solution found at step:	**108491336**
Objective value:	**9.000000**
Branch count:	**635940**

Variable	Value	Reduced Cost
AIRCRAFTNUMBER(1)	1.000000	1.000000
AIRCRAFTNUMBER(2)	1.000000	1.000000
AIRCRAFTNUMBER(3)	1.000000	1.000000
AIRCRAFTNUMBER(4)	0.0000000	1.000000
AIRCRAFTNUMBER(5)	1.000000	1.000000
AIRCRAFTNUMBER(6)	1.000000	1.000000
AIRCRAFTNUMBER(7)	1.000000	1.000000
AIRCRAFTNUMBER(8)	1.000000	1.000000
AIRCRAFTNUMBER(9)	1.000000	1.000000
MISSIONLENGTH(1)	495.0000	0.0000000
MISSIONLENGTH(2)	810.0000	0.0000000
MISSIONLENGTH(3)	735.0000	0.0000000
MISSIONLENGTH(4)	435.0000	0.0000000
MISSIONLENGTH(5)	855.0000	0.0000000
MISSIONLENGTH(6)	555.0000	0.0000000
MISSIONLENGTH(7)	495.0000	0.0000000
MISSIONLENGTH(8)	140.0000	0.0000000
MISSIONLENGTH(9)	140.0000	0.0000000
MISSIONLENGTH(10)	140.0000	0.0000000
MISSIONLENGTH(11)	210.0000	0.0000000
MISSIONLENGTH(12)	210.0000	0.0000000
MISSIONLENGTH(13)	210.0000	0.0000000
MISSIONLENGTH(14)	210.0000	0.0000000
MISSIONLENGTH(15)	210.0000	0.0000000
MISSIONLENGTH(16)	210.0000	0.0000000
MISSIONLENGTH(17)	195.0000	0.0000000
MISSIONLENGTH(18)	195.0000	0.0000000
MISSIONLENGTH(19)	195.0000	0.0000000
MISSIONLENGTH(20)	195.0000	0.0000000
MISSIONLENGTH(21)	195.0000	0.0000000
MISSIONLENGTH(22)	195.0000	0.0000000
MISSIONLENGTH(23)	105.0000	0.0000000
MISSIONLENGTH(24)	105.0000	0.0000000
MISSIONLENGTH(25)	160.0000	0.0000000
MISSIONLENGTH(26)	160.0000	0.0000000
MISSIONLENGTH(27)	160.0000	0.0000000
MISSIONLENGTH(28)	255.0000	0.0000000

MISSIONLENGTH(29)	255.0000	0.0000000
MISSIONLENGTH(30)	255.0000	0.0000000
MISSIONLENGTH(31)	255.0000	0.0000000
MISSIONLENGTH(32)	255.0000	0.0000000
MISSIONLENGTH(33)	225.0000	0.0000000
MISSIONLENGTH(34)	225.0000	0.0000000
MISSIONLENGTH(35)	225.0000	0.0000000
MISSIONLENGTH(36)	225.0000	0.0000000
MISSIONLENGTH(37)	160.0000	0.0000000
MISSIONLENGTH(38)	160.0000	0.0000000
MISSIONLENGTH(39)	160.0000	0.0000000
MISSIONLENGTH(40)	160.0000	0.0000000
MISSIONLENGTH(41)	160.0000	0.0000000
MISSIONLENGTH(42)	160.0000	0.0000000
MISSIONLENGTH(43)	330.0000	0.0000000
MISSIONLENGTH(44)	258.0000	0.0000000
MISSIONLENGTH(45)	294.0000	0.0000000
MISSIONLENGTH(46)	288.0000	0.0000000
MISSIONLENGTH(47)	342.0000	0.0000000
MISSIONLENGTH(48)	228.0000	0.0000000
MISSIONLENGTH(49)	204.0000	0.0000000
MISSIONLENGTH(50)	306.0000	0.0000000
ASSIGN(1, 3)	1.000000	0.0000000
ASSIGN(2, 6)	1.000000	0.0000000
ASSIGN(3, 8)	1.000000	0.0000000
ASSIGN(4, 2)	1.000000	0.0000000
ASSIGN(5, 8)	1.000000	0.0000000
ASSIGN(6, 9)	1.000000	0.0000000
ASSIGN(7, 2)	1.000000	0.0000000
ASSIGN(8, 1)	1.000000	0.0000000
ASSIGN(9, 5)	1.000000	0.0000000
ASSIGN(10, 1)	1.000000	0.0000000
ASSIGN(11, 1)	1.000000	0.0000000
ASSIGN(12, 1)	1.000000	0.0000000
ASSIGN(13, 1)	1.000000	0.0000000
ASSIGN(14, 1)	1.000000	0.0000000
ASSIGN(15, 1)	1.000000	0.0000000
ASSIGN(16, 5)	1.000000	0.0000000
ASSIGN(17, 1)	1.000000	0.0000000
ASSIGN(18, 1)	1.000000	0.0000000
ASSIGN(19, 7)	1.000000	0.0000000
ASSIGN(20, 6)	1.000000	0.0000000
ASSIGN(21, 6)	1.000000	0.0000000
ASSIGN(22, 6)	1.000000	0.0000000
ASSIGN(23, 6)	1.000000	0.0000000
ASSIGN(24, 7)	1.000000	0.0000000
ASSIGN(25, 7)	1.000000	0.0000000
ASSIGN(26, 7)	1.000000	0.0000000
ASSIGN(27, 8)	1.000000	0.0000000
ASSIGN(28, 2)	1.000000	0.0000000
ASSIGN(29, 2)	1.000000	0.0000000
ASSIGN(30, 2)	1.000000	0.0000000
ASSIGN(31, 9)	1.000000	0.0000000
ASSIGN(32, 3)	1.000000	0.0000000
ASSIGN(33, 3)	1.000000	0.0000000
ASSIGN(34, 3)	1.000000	0.0000000
ASSIGN(35, 7)	1.000000	0.0000000

ASSIGN(36, 7)	1.000000	0.0000000
ASSIGN(37, 7)	1.000000	0.0000000
ASSIGN(38, 7)	1.000000	0.0000000
ASSIGN(39, 7)	1.000000	0.0000000
ASSIGN(40, 5)	1.000000	0.0000000
ASSIGN(41, 5)	1.000000	0.0000000
ASSIGN(42, 5)	1.000000	0.0000000
ASSIGN(43, 5)	1.000000	0.0000000
ASSIGN(44, 6)	1.000000	0.0000000
ASSIGN(45, 9)	1.000000	0.0000000
ASSIGN(46, 9)	1.000000	0.0000000
ASSIGN(47, 9)	1.000000	0.0000000
ASSIGN(48, 3)	1.000000	0.0000000
ASSIGN(49, 7)	1.000000	0.0000000
ASSIGN(50, 3)	1.000000	0.0000000

Solution (With max. number of missions)

Global optimal solution found at step:	**1894834985**
Objective value:	**10.000000**
Branch count:	**980960**

Variable	Value	Reduced Cost
AIRCRAFTNUMBER(1)	1.000000	1.000000
AIRCRAFTNUMBER(2)	1.000000	1.000000
AIRCRAFTNUMBER(3)	1.000000	1.000000
AIRCRAFTNUMBER(4)	1.000000	1.000000
AIRCRAFTNUMBER(5)	1.000000	1.000000
AIRCRAFTNUMBER(6)	1.000000	1.000000
AIRCRAFTNUMBER(7)	1.000000	1.000000
AIRCRAFTNUMBER(8)	1.000000	1.000000
AIRCRAFTNUMBER(9)	1.000000	1.000000
AIRCRAFTNUMBER(10)	1.000000	1.000000
MISSIONLENGTH(1)	495.0000	0.0000000
MISSIONLENGTH(2)	810.0000	0.0000000
MISSIONLENGTH(3)	735.0000	0.0000000
MISSIONLENGTH(4)	435.0000	0.0000000
MISSIONLENGTH(5)	855.0000	0.0000000
MISSIONLENGTH(6)	555.0000	0.0000000
MISSIONLENGTH(7)	495.0000	0.0000000
MISSIONLENGTH(8)	140.0000	0.0000000
MISSIONLENGTH(9)	140.0000	0.0000000
MISSIONLENGTH(10)	140.0000	0.0000000
MISSIONLENGTH(11)	140.0000	0.0000000
MISSIONLENGTH(12)	140.0000	0.0000000
MISSIONLENGTH(13)	140.0000	0.0000000
MISSIONLENGTH(14)	140.0000	0.0000000
MISSIONLENGTH(15)	210.0000	0.0000000
MISSIONLENGTH(16)	210.0000	0.0000000
MISSIONLENGTH(17)	210.0000	0.0000000
MISSIONLENGTH(18)	210.0000	0.0000000
MISSIONLENGTH(19)	210.0000	0.0000000
MISSIONLENGTH(20)	210.0000	0.0000000
MISSIONLENGTH(21)	210.0000	0.0000000
MISSIONLENGTH(22)	195.0000	0.0000000
MISSIONLENGTH(23)	195.0000	0.0000000
MISSIONLENGTH(24)	195.0000	0.0000000
MISSIONLENGTH(25)	195.0000	0.0000000
MISSIONLENGTH(26)	195.0000	0.0000000
MISSIONLENGTH(27)	195.0000	0.0000000

MISSIONLENGTH(28)	195.0000	0.0000000
MISSIONLENGTH(29)	195.0000	0.0000000
MISSIONLENGTH(30)	195.0000	0.0000000
MISSIONLENGTH(31)	195.0000	0.0000000
MISSIONLENGTH(32)	105.0000	0.0000000
MISSIONLENGTH(33)	105.0000	0.0000000
MISSIONLENGTH(34)	105.0000	0.0000000
MISSIONLENGTH(35)	160.0000	0.0000000
MISSIONLENGTH(36)	160.0000	0.0000000
MISSIONLENGTH(37)	160.0000	0.0000000
MISSIONLENGTH(38)	160.0000	0.0000000
MISSIONLENGTH(39)	160.0000	0.0000000
MISSIONLENGTH(40)	255.0000	0.0000000
MISSIONLENGTH(41)	255.0000	0.0000000
MISSIONLENGTH(42)	255.0000	0.0000000
MISSIONLENGTH(43)	255.0000	0.0000000
MISSIONLENGTH(44)	255.0000	0.0000000
MISSIONLENGTH(45)	255.0000	0.0000000
MISSIONLENGTH(46)	255.0000	0.0000000
MISSIONLENGTH(47)	225.0000	0.0000000
MISSIONLENGTH(48)	225.0000	0.0000000
MISSIONLENGTH(49)	225.0000	0.0000000
MISSIONLENGTH(50)	225.0000	0.0000000
MISSIONLENGTH(51)	225.0000	0.0000000
MISSIONLENGTH(52)	225.0000	0.0000000
MISSIONLENGTH(53)	160.0000	0.0000000
MISSIONLENGTH(54)	160.0000	0.0000000
MISSIONLENGTH(55)	160.0000	0.0000000
MISSIONLENGTH(56)	160.0000	0.0000000
MISSIONLENGTH(57)	160.0000	0.0000000
MISSIONLENGTH(58)	160.0000	0.0000000
MISSIONLENGTH(59)	160.0000	0.0000000
MISSIONLENGTH(60)	160.0000	0.0000000
MISSIONLENGTH(61)	160.0000	0.0000000
MISSIONLENGTH(62)	160.0000	0.0000000
MISSIONLENGTH(63)	330.0000	0.0000000
MISSIONLENGTH(64)	258.0000	0.0000000
MISSIONLENGTH(65)	294.0000	0.0000000
MISSIONLENGTH(66)	288.0000	0.0000000
MISSIONLENGTH(67)	342.0000	0.0000000
MISSIONLENGTH(68)	228.0000	0.0000000
MISSIONLENGTH(69)	204.0000	0.0000000
MISSIONLENGTH(70)	306.0000	0.0000000
ASSIGN(1, 2)	1.000000	0.0000000
ASSIGN(2, 6)	1.000000	0.0000000
ASSIGN(3, 9)	1.000000	0.0000000
ASSIGN(4, 5)	1.000000	0.0000000
ASSIGN(5, 7)	1.000000	0.0000000
ASSIGN(6, 1)	1.000000	0.0000000
ASSIGN(7, 1)	1.000000	0.0000000
ASSIGN(8, 1)	1.000000	0.0000000
ASSIGN(9, 3)	1.000000	0.0000000
ASSIGN(10, 7)	1.000000	0.0000000
ASSIGN(11, 7)	1.000000	0.0000000
ASSIGN(12, 7)	1.000000	0.0000000
ASSIGN(13, 7)	1.000000	0.0000000
ASSIGN(14, 5)	1.000000	0.0000000

ASSIGN(15, 9)	1.000000	0.0000000
ASSIGN(16, 9)	1.000000	0.0000000
ASSIGN(17, 2)	1.000000	0.0000000
ASSIGN(18, 9)	1.000000	0.0000000
ASSIGN(19, 2)	1.000000	0.0000000
ASSIGN(20, 8)	1.000000	0.0000000
ASSIGN(21, 2)	1.000000	0.0000000
ASSIGN(22, 5)	1.000000	0.0000000
ASSIGN(23, 4)	1.000000	0.0000000
ASSIGN(24, 4)	1.000000	0.0000000
ASSIGN(25, 4)	1.000000	0.0000000
ASSIGN(26, 4)	1.000000	0.0000000
ASSIGN(27, 3)	1.000000	0.0000000
ASSIGN(28, 6)	1.000000	0.0000000
ASSIGN(29, 1)	1.000000	0.0000000
ASSIGN(30, 5)	1.000000	0.0000000
ASSIGN(31, 10)	1.000000	0.0000000
ASSIGN(32, 7)	1.000000	0.0000000
ASSIGN(33, 7)	1.000000	0.0000000
ASSIGN(34, 7)	1.000000	0.0000000
ASSIGN(35, 5)	1.000000	0.0000000
ASSIGN(36, 8)	1.000000	0.0000000
ASSIGN(37, 4)	1.000000	0.0000000
ASSIGN(38, 4)	1.000000	0.0000000
ASSIGN(39, 5)	1.000000	0.0000000
ASSIGN(40, 8)	1.000000	0.0000000
ASSIGN(41, 2)	1.000000	0.0000000
ASSIGN(42, 9)	1.000000	0.0000000
ASSIGN(43, 4)	1.000000	0.0000000
ASSIGN(44, 8)	1.000000	0.0000000
ASSIGN(45, 5)	1.000000	0.0000000
ASSIGN(46, 1)	1.000000	0.0000000
ASSIGN(47, 8)	1.000000	0.0000000
ASSIGN(48, 8)	1.000000	0.0000000
ASSIGN(49, 8)	1.000000	0.0000000
ASSIGN(50, 2)	1.000000	0.0000000
ASSIGN(51, 3)	1.000000	0.0000000
ASSIGN(52, 3)	1.000000	0.0000000
ASSIGN(53, 3)	1.000000	0.0000000
ASSIGN(54, 2)	1.000000	0.0000000
ASSIGN(55, 5)	1.000000	0.0000000
ASSIGN(56, 1)	1.000000	0.0000000
ASSIGN(57, 3)	1.000000	0.0000000
ASSIGN(58, 3)	1.000000	0.0000000
ASSIGN(59, 3)	1.000000	0.0000000
ASSIGN(60, 9)	1.000000	0.0000000
ASSIGN(61, 6)	1.000000	0.0000000
ASSIGN(62, 6)	1.000000	0.0000000
ASSIGN(63, 10)	1.000000	0.0000000
ASSIGN(64, 10)	1.000000	0.0000000
ASSIGN(65, 10)	1.000000	0.0000000
ASSIGN(66, 6)	1.000000	0.0000000
ASSIGN(67, 10)	1.000000	0.0000000
ASSIGN(68, 3)	1.000000	0.0000000
ASSIGN(69, 8)	1.000000	0.0000000
ASSIGN(70, 10)	1.000000	0.0000000

BIBLIOGRAPHY

Baker, K.R. Introducing to Sequencing and Scheduling. Wiley, New York, 1974.

Bazaraa , Mokhtar S., John J. Jarvis, Hanif D. Sherali. Linear Programming and Network Flows. Courier Companies, Inc, 1997.

Del Rosario, Michael. Scheduling Air Mobility Command's Channel Cargo Missions. MS Thesis , Wright- Patterson AFB OH, 1993.

Discover Turkey. Geographic Map of Turkey. 19 January 2000. (http://www.turkishnews.com/DiscoverTurkey/images/maps/map1.jpg)

Embassy of the Republic of Turkey, Washington, DC. Country Profile: The Turkish Reference Library. 19 January, 2000. (http://turkey.org/reference_fr.html)

Embassy of the Republic of Turkey, Washington, DC. Politics and Policy: The Turkish Politics and Policy. 19 January 2000. (http://turkey.org/reference_fr.html)

French, S. Sequencing and Scheduling.Horwood, Chichester, 1982.

Lingo. User Manual. Belment, Wadsworth Publishing Company, 1997.

Morton, T.E and Pentico. Heuristic Scheduling Systems. Wiley, NewYork, 1993.

Moul, Justin Edward. A method for Determining Schedule Delay Information in a Channel Cargo Route Network Schedule. MS Thesis, Air Force Institute of Technology, 1992.

Pinedo, Micheal. Scheduling Theory, Algorithms and System. Prentice Hall Inc, 1995.

Republic of Turkey Ministry of Foreign Affairs. Foreign Policy: The Goals and Principles of Turkish Foreign Policy. 19 January 2000. (www.access.ch/tuerkei/GRUPF/princip.htm#Global)

Republic of Turkey Ministry of Foreign Affairs. Foreign Policy: The Relations with US. 19 January 2000. (www.access.ch/tuerkei/GRUPF/princip.htm#Global)

Smith, Douglas R., Eduardo A.Parra, and Stephen H. Westfold. Generic Tools for Transportation Planning and Scheduling. Griffiss Air force Base, N. Y., 1995.

Smith, R. Douglas and Stephen H. Westfold. Generic Tools For Transportation Planning And Scheduling: Technical Report, 07 Mar 1991 – 30 Sep 94. Contract F30602-91-C-0043. Kestrel Institute, August 1995.

Topcuoglu, H. Turkish Air Mobility Modeling. MS Thesis, Wright-Patterson AFB OH, 1997.

Winston, Wayne W. Operation Research Application and Algorithms. Third edition. Duxburg Press, 1994.

VITA

First Lieutenant Mehmet M. Arı was born on 14 May 1972, in Şanliurfa, Türkiye. He graduated from Istanbul High School in 1990. He got his education in German in Istanbul High School. He entered the Turkish Air Force Academy. He received the degree of Bachelor of Science in Electronics in August 1994. Upon graduation he attended pilot training course and became a pilot in 1996. Then he was assigned to 172nd Squadron as a F – 4 E Phantom pilot after F – 4 E training course. Having served one and half year in 172nd Squadron, he entered the Graduate School of Engineering, Air Force Institute of Technology in August 1998. He is awarded being a member of Tau Beta Pi (TBP-The National Engineering Honor Society) in AFIT. He speaks German and English.

Permanent Address: Oğuzhan cad. No: 12.

D. 6 Findikzade/Istanbul

Turkiye

E-mail: Mehmetari94@hotmail.com